Human Machine Collaborative Decision Making in a Complex Optimization System

by

Jeremy S. Malasky

B.S. Operations Research
United States Air Force Academy, 2003

SUBMITTED TO THE SLOAN SCHOOL OF MANAGEMENT
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF

MASTERS OF SCIENCE IN OPERATIONS RESEARCH
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005

Signature of Author: ____

Sloan School of Management
Interdepartmental Program in Operations Research
12 May, 2005

Approved by: ____

Milton B. Adams
Director, Strategic Technology Planning
The Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Certified by: ____

Cynthia Barnhart
Professor, Civil and Environmental Engineering and Engineering Systems Division
Thesis Advisor

Accepted by: ____

James B. Orlin
Edward Pennell Brooks Professor of Operations Research
Co-Director, Operations Research Center

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **00 JUN 2005**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**
4. TITLE AND SUBTITLE **Human Machine Collaborative Decision Making in a Complex Optimization System**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Massachusetts Institute Of Technology**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**151**	

[This Page Intentionally Left Blank]

Human Machine Collaborative Decision Making in a Complex Optimization System

by

Jeremy S. Malasky

Submitted to the Sloan School of Management
On 12 May 2005, in Partial Fulfillment of the Requirements
For the Degree of Masters of Science in Operations Research

ABSTRACT

Numerous complex real-world applications are either theoretically intractable or unable to be solved in a practical amount of time. Researchers and practitioners are forced to implement heuristics in solving such problems that can lead to highly sub-optimal solutions. Our research focuses on inserting a human "in-the-loop" of the decision-making or problem solving process in order to generate solutions in a timely manner that improve upon those that are generated either solely by a human or solely by a computer. We refer to this as *Human-Machine Collaborative Decision-Making (HMCDM)*.

The typical design process for developing human-machine approaches either starts with a human approach and augments it with decision-support or starts with an automated approach and augments it with operator input. We provide an alternative design process by presenting an HMCDM methodology that addresses collaboration from the outset of the design of the decision-making approach.

We apply this design process to a complex military resource allocation and planning problem which *selects*, *sequences*, and *schedules* teams of unmanned aerial vehicles (UAVs) to perform sensing (Intelligence, Surveillance, and Reconnaissance – ISR) and strike activities against enemy targets. Specifically, we examined varying degrees of human-machine collaboration in the creation of variables in the solution of this problem. We also introduce an HMCDM method that combines traditional goal decomposition with a model formulation into an *Iterative Composite Variable Approach* for solving large-scale optimization problems. Finally, we show through experimentation the potential for improvement in the quality and speed of solutions that can be achieved through the use of an HMCDM approach.

Thesis Supervisor: Professor Cynthia Barnhart
Title: Professor of Civil and Environmental Engineering and Engineering Systems

Technical Supervisor: Dr. Milton B. Adams
Title: Director, Strategic Technology Planning,
The Charles Stark Draper Laboratory, Inc.

[This Page Intentionally Left Blank]

Acknowledgements

There are many to whom I owe thanks for my experience at MIT. The first is Milt Adams of Draper Laboratory. His dedication, expertise and guidance have truly made all the difference in my research.

I am equally thankful for Professor Cynthia Barnhart of MIT for always finding time to provide her knowledge no matter how hectic and busy her schedule.

I must also thank Ramsay Key, Alex Kahn, and Laura Major Forest. I couldn't have asked for a better team of professionals to work with. Their insight was truly invaluable to the progression of my research.

I would like to thank William Kreamer and Chris Guerra for the endless time spent initially familiarizing me with the massive amounts of code in MICA.

Thanks to Jeff Cipollini. His dedication to the C4I Lab and words of wisdom certainly enriched my research experience.

I am grateful to the Charles Stark Draper Laboratory, the Massachusetts Institute of Technology, and the Operations Research Center for providing me with this amazing opportunity, and to the United States Air Force for allowing me to pursue it.

I am equally grateful to Andrew Armacost of the Air Force Academy for encouraging and assisting me in pursuing this invaluable educational opportunity.

Last but certainly not least, I am especially grateful for the love, support, and encouragement my family and friends have provided me through all of my endeavors including my stay here at MIT.

Publication of this thesis does not constitute approval by Draper of the findings or conclusions herein. It is published for the exchange and stimulation of ideas.

Finally, as a member of the Air Force, I acknowledge that the views expressed in this thesis are mine and do not reflect official policy or position of the United States Air Force, the Department of the Defense, or the United States Government.

Jeremy S. Malasky
2nd Lieutenant, USAF

Assignment

Draper Laboratory Report Number T-1513

In consideration for the research opportunity and permission to prepare my thesis by and at The Charles Stark Draper Laboratory, Inc., I hereby assign my copyright of the thesis to The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts.

____

Jeremy S. Malasky, 2nd Lieutenant, USAF Date

Table of Contents

[This Page Intentionally Left Blank]

List of Figures

List of Tables

[This Page Intentionally Left Blank]

Chapter 1

Introduction

1.1 Problem Statement

Even with increases in computational power, advances in algorithms, and the development of more complex modeling capabilities, there remain numerous applications that are either theoretically intractable or unable to be solved in a practical amount of time. When attacking such problems, researchers and practitioners are forced to implement heuristics. This approach has a significant limitation in that the heuristics invariably lead to sub-optimal solutions. An alternative to employing this approach, or as an augmentation to this approach, is to analyze problems and look for areas to insert a human "in-the-loop" of the decision-making or problem solving process. The objective is to decide how to allocate decisions intelligently and at what level of automation [29] to place these decisions. The goal of this approach is to combine the intuition and experience of a human with the computational speed of a computer system in order to generate solutions in a timely manner that improve upon those that are generated either solely by a human or a computer. Improved solutions include those with more "value" or those with the same value but which are generated more quickly. We refer to this process as *Human-Machine Collaborative Decision-Making (HMCDM).*

1.2 Motivation

There are four classes of problems which we believe will benefit from HMCDM. We refer to the first class of problems as *combinatorial problems.* These are problems that are impractical to solve to optimality due to a sufficiently large search space. That is, the size of the search space is such that it prevents the problem from ever being solved to optimality or that it requires an unacceptably long amount of time for the computer to exhaustively search the space. In this case, a human might be able to assist by narrowing or focusing the search space; a process similar to pruning a decision tree.

We refer to the second class of problems as *visual problems.* Visual problems are those that take an inherently visual form and those that are abstract but could ultimately be represented visually. Some examples of inherently visual problems include geographical clustering, image classification, regression or curve fitting, and small graph problems. An example of an abstract problem that may be represented visually would be a problem that is solved by an iterative process in order to arrive at a final solution. As a by-product of the iterative process, there is a solution or solutions generated at each step. Presenting to the human operator an appropriate visual representation of the current solution(s) at each iteration might enable the operator to guide the process toward optimality or choose certain pieces of the solution to "hold onto" for the next iteration.

Problems in which a human monitors the amount of machine computation effort could also benefit from HMCDM. We refer to this third class of problems as *computationally intensive problems.* The human could monitor the process, identify a point of diminishing return and stop the computer from searching for new solutions. The thought is that humans have good intuition when weighing the cost of further computation versus the potential benefit of this added computation to the overall solution value. The ability of a human to control the computational effort dynamically might prove more effective than a fixed strategy.

Finally, by applying HMCDM, problems whose solution approach employs a large number of different heuristics stand to benefit as well. We refer to this fourth class of problems as *heuristic-heavy problems.* Because the use of different heuristics can lead to different solutions, a human can choose which heuristic to select and when to select it. A more complex approach might allow the operator to adapt heuristics dynamically as a function of the problem or during the evolution of the solution (see the discussion of iterative approaches above). This may result in better solutions than if a fixed approach to applying heuristics were used.

In addition to applying HMCDM to problems in the hope of generating "better" solutions, it is necessary to also employ such an approach in problems that may already be solved completely with the sole use of a computer or a machine. One key to obtaining usable and effective solutions is that system users or program operators understand and trust the generated solutions; neither of which is guaranteed with solutions created entirely by a computer. If two identical solutions are generated for a particular problem, one created exclusively by a computer and one created via human-machine collaboration, a human will be more likely to accept the solution in which they were involved in the decision making process [20] [26] [27].

1.3 Thesis Problem

For this thesis, we will study the application of HMCDM for a military command and control (C^2) system for resource allocation and planning. The experiments conducted for this research build on software developed previously for the Defense Advanced Research Projects Agency (DARPA) Mixed-Initiative Control of Automa-teams (MICA) program designed specifically to simulate a C^2 system of resource allocation and planning. The MICA solution is a closed-loop, dynamic planning and execution system intended to aid a human in making decisions about courses of action related to military planning. The initial inputs into MICA are a list of assets, resources associated with those assets, enemy targets, and Commander's Intent. Based on this information, the goal is to *select, sequence,* and *schedule* sensing (Intelligence, Surveillance, and Reconnaissance – ISR) and strike activities for the available aircraft resources to address enemy targets in an effort to maximize the total expected value minus cost.

This application contains every one of the classes of problems for which HMCDM might be helpful. First, this problem is extremely complex and, thus, difficult to solve. One reason for the difficulty is the large search space resulting from the vast number of decision variables. Another is the fact that many of these variables change dynamically. In addition to variables changing over time, there are many probabilistic aspects of the problem. For example, there is uncertainty about the location of the enemy targets, identification of targets, effectiveness of weapons used on particular targets, and the damage state of targets. This makes it impossible to enumerate all possible decision variables and outcomes and then apply some computer algorithm to solve the problem. The problem also lends itself to being naturally portrayed in a visual manner. The geographical layout of the scenarios that include the enemy targets and friendly resources is easily represented in a map-based Graphical User Interface. In addition, the MICA problem could benefit from having a human control the amount of computation time spent on various subproblems throughout the system. The thought is that a human can effectively manage the computational effort expended on problems. Research at Mitsubishi Electric Research Labs [4] [27] has shown that humans can successfully weigh the cost of further computation versus the potential benefit of this computation. Finally, the MICA problem contains a large number of heuristics. The purpose of the system is to plan missions involving numerous friendly resources and enemy targets in real-time. This large and dynamically changing problem is therefore more easily solved using heuristics instead of other techniques. The application in this thesis deals with determining how humans and computer optimization algorithms can complement each other to provide viable solutions in such time critical resource allocation and planning scenario.

1.4 Contributions

In order to appreciate the benefit of having humans and machines collaborate together when solving optimization problems, it is first necessary to understand the traditional human-machine decision making interaction. In general, there have been two approaches to human-machine problem solving; we will refer to these as the human factors approach and the algorithmic or optimization approach. In the human factors approach, systems are designed from a human's perspective in that their main focus is on the human while they attempt to use computer technology or automation to augment, mimic or enhance the human's approach to solving the problem. On the other hand, the algorithmic or optimization approach typically focuses on an algorithmic approach to solving the problem – one that is, at least initially, developed without consideration of human participation in problem solving. Thus, algorithmic approaches attempt to model as much as possible and assign whatever remains to the human operator. There has been limited work in trying to exploit the strengths of both the computer and human from the outset.

When algorithmic approaches are applied to optimization problems in particular, the human involvement is typically very limited. In the design phase, a human (problem formulator and algorithmic designer) is tasked with understanding the physical constraints and objective and translating these into mathematical equations and lines of computer code. A human might also be involved in the actual operational phase, but again in a limited manner. Typically, this involvement is limited to a user inputting some initial data or parameters and then allowing the computer to solve the problem.

It is remarkable that there has been little work done in both the general HMCDM area and more specifically in applying HMCDM to optimization problems. This thesis contributes to filling that void in the following ways:

- Typical human-machine approaches start with a human process and augment it with decision-support, or start with an automated process and augment it with operator input. We provide an alternative to these approaches by presenting an HMCDM methodology that addresses collaboration from the outset of the decision-making design process.
- We apply this approach to a complex military resource allocation and planning problem and show through experimentation the potential for improvement in the quality and speed of solutions.
- We update and build upon previously accepted lists of human and computer strengths and capabilities.
- We build upon previous research to propose a methodology for determining the optimal level of automation when allocating decisions in a system or algorithm.
- We introduce a method for combining traditional goal decomposition [10] [3] with composite variable formulation [5] into an *Iterative Composite Variable Approach* for solving large-scale optimization problems.

1.5 *Thesis Overview and Content*

The individual chapters are summarized as follows:

Chapter 2: *Previous HMCDM Research*

In this chapter we provide a review of previous research conducted in the realm of Human-Machine Collaborative Decision Making. A focus of this chapter is Sheridan and Verplank's 10 levels of human-machine automation [29]. In addition, the chapter outlines research pertaining to human-machine decision allocation which addresses who (human or machine) should be making which decisions throughout a system or problem solving process. We also update and build upon the currently accepted human and computer respective strengths and capabilities. This chapter ends with our proposed method for determining not only who should be making which decisions throughout a problem solving process but also at what level of automation these decisions should be made. A brief example is given at the conclusion of this chapter in which we apply our methodology to the resource allocation and planning problem which we will use later in the thesis as our test-bed for human-machine interaction.

Chapter 3: *MICA System*

The purpose of this chapter is to provide an overview of the Mixed-Initiative Control of Automa-teams (MICA) system which is the platform for our human-machine collaboration experiments. We provide background information as to why the system was created as well as a breakdown of the three-tiered hierarchical decomposition planning algorithm it employs. In addition, we introduce the concept of *composite variables* which was presented in [5] and will be explored in further detail in Chapter 4. We conclude the chapter by outlining particular subproblems within the MICA system that might benefit the most from applying HMCDM.

Chapter 4: *Large Scale Optimization – Goal Decomposition & Composite Variable Formulation*

Two popular techniques for solving large scale optimization problems such as the MICA C^2 problem are *goal decomposition* and *composite variable formulation.* This chapter explains both methods as well as describes their similarities and differences. We discuss their respective strengths and weaknesses in the context of addressing complex large-scale optimization problems. Finally, we outline a proposed strategy for incorporating both methods in an HMCDM context. We call this strategy the *Iterative Composite Variable Approach*. We end the chapter by describing how this approach can be used in the MICA application to generate "better" solutions.

Chapter 5: *Setup of Experiments*

In this chapter we outline the setup of the MICA HMCDM experiments along with our goals and hypotheses. We introduce the concept of *Key Pieces of Information (KPI)*, information that is generated by the computer for the human to use in aiding their task of creating clusters of enemy targets. We provide a flow of the experiment along with visual images from the MICA system to elucidate the process through which our experiment subjects proceeded. This chapter also details the metrics that are used for evaluating human involvement.

Chapter 6: *Results of Experiments*

We present an explanation and rationale for each of the five scenarios used in the MICA HMCDM experiments. This chapter also includes all data output from the respective scenario experiments. We provide analysis of the HMCDM experiments and discuss the benefits of human-machine collaboration over both 'computer only' and 'human only' approaches.

Chapter 7: *Summary and Future Work*

We conclude the thesis with a chapter summarizing both the general concept of *Human Machine Collaborative Decision Making* as well as the empirical results obtained from the HMCDM MICA experiment. We also discuss future research in this chapter.

Chapter 2

HMCDM Review

This chapter reviews previous research in the field of Human-Machine Collaborative Decision Making (HMCDM). Various techniques for determining the optimal allocation of human-machine decision-making are outlined. We also review and augment existing characterizations of human and computer strengths and capabilities. Another significant area of research covered in this chapter is Sheridan and Verplank's 10 Levels of Automation of Decision and Action Selection [29]. We also augment Sheridan, Parasuraman, and Wicken's list of *Evaluative Criteria* [28] - factors that are considered when determining the level of automation that should be employed in executing a certain task or making a certain decision.

The limited existing research pertaining to the application of HMCDM to optimization problems is highlighted in this chapter. The chapter ends with the description of a *Level of Automation* methodology developed during the course of this thesis research for determining not only who[1] should be making which decisions but also at what level of automation these decisions should be made. The chapter concludes with a brief example of the application of our methodology to the resource allocation and planning problem described in Chapter 3 and used during the course of this thesis research.

2.1 Previous HMCDM Research

Typical approaches to the design of human-machine collaboration either start with a human process of problem solving and augment it with decision-support, or start with an

[1] Throughout the thesis, "who is making the decision" refers to either a human or a machine

automated process and augment it with operator input. There has been limited research addressing human-machine collaboration that has a starting point that is neither a human-centered nor a machine-centered approach to the decision making process. Our hypothesis is that such an approach ultimately allows for more tightly coupled and synergistic interaction between human and machine. To design such a system, an intelligent choice as to which decisions and actions might be allocated to human operators and which might be allocated to computer resources must be made at the outset. Note that this allocation is not meant to be exclusive in that there is likely to be a subset of decisions that might be appropriate for either operator or machine. It is that subset that presents the significant design challenge. That is, when (i.e., under what circumstances) should those decisions be allocated to operator alone, to machine alone or to a collaborative effort of operator and machine. A variety of approaches have been developed for determining the “proper” allocation of decisions including: ad hoc approaches [18] , formal approaches which include the comparative assessment of human and machine performance using qualitative listings [18] [15] [32], balanced approaches which are a combination of ad hoc and formal [22], two-dimensional capability scaling graphs [25], and varying levels of human-machine collaboration [29]. Each of these is discussed in more detail in the following sections.

2.2 Background on Allocation Approaches

2.2.1 Ad Hoc Decision Allocation Approach

The first approach, referred to as the "ad hoc" or “gut feel” approach, assumes that the decision allocations in existing systems are satisfactory, and that only minor changes to their level of automation are required for improved performance. Although hypothetically changes could be made to either increase or decrease the level of automation, typically changes are made to increase the computer involvement in such ad hoc systems. The decision or action allocations are based on the economically available level of automation, and are made almost entirely on criteria such as cost, availability, reliability, and compatibility of hardware and software. Decisions in the existing system are examined in order to determine if changes should be made as to who is making the decision. However, no thorough analysis has been conducted in order to come to this conclusion. Rather, it is determined solely on a “gut feel” or “trial and error” basis.

The advantage of such an approach is that it is simple, involves minimum effort and is reported to be low cost [18]. Unfortunately, this approach has numerous disadvantages. The first is that such an approach lacks standardization. Different opinions by separate managers, designers, or operators may result in extremely different outcomes as to who is given control over certain decisions. The typical outcome of such an approach is that decisions are allocated to the machine on the basis of what can be done by machine, leaving the human to perform whatever is left that cannot be done by machine (at least cost effectively). This poses a potential problem in that the "left over" tasks may not form a coherent set. Finally, the analysis typically addresses one decision at a time and does not necessarily evaluate how that decision will affect other interconnected decisions throughout the system. Thus, the ad hoc approach is likely to be ineffective for complex decision making systems.

2.2.2 Formal and Balanced Decision Allocation Approaches

An alternative to the ad hoc approach has been called the "formal approach" [18]. This formally allocates each system decision to either a human or a machine using a rational decision making technique. Although the formal technique does not ensure the optimum allocation of decisions, it goes beyond the informal, or "gut feel," method which is so often used. More detail of formal approaches will be outlined in Section 2.3.

Neither the ad-hoc nor the formal approaches are strictly followed in practice: what usually happens is referred to as a “balanced approach” which is a combination of both the ad hoc and formal approaches [18]. This approach accounts for political, managerial, and performance constraints on certain decisions that are to be made. For example, some decisions must be assigned to humans for political, legal or doctrinal reasons (e.g., the military deciding to release certain types of weapons). Furthermore, some decision allocations may be dictated by performance requirements, such as the need to respond in a limited time, the need to maintain operator skills, or the space and weight constraints associated with accommodating human operators. The ad hoc approach can be used to ensure such political, financial, managerial, and performance constraints. This leaves a much smaller set of decisions or actions to be addressed by a formal analysis. Meister [22] outlines the balanced approach as the five stages, shown in Figure 2-1.

The balanced approach is a better reflection of how decisions are typically assigned on major projects than either the ad hoc or the formal approaches. Note that Meister's outline calls for the use of formal techniques to be used in the third and fourth steps of his algorithm.

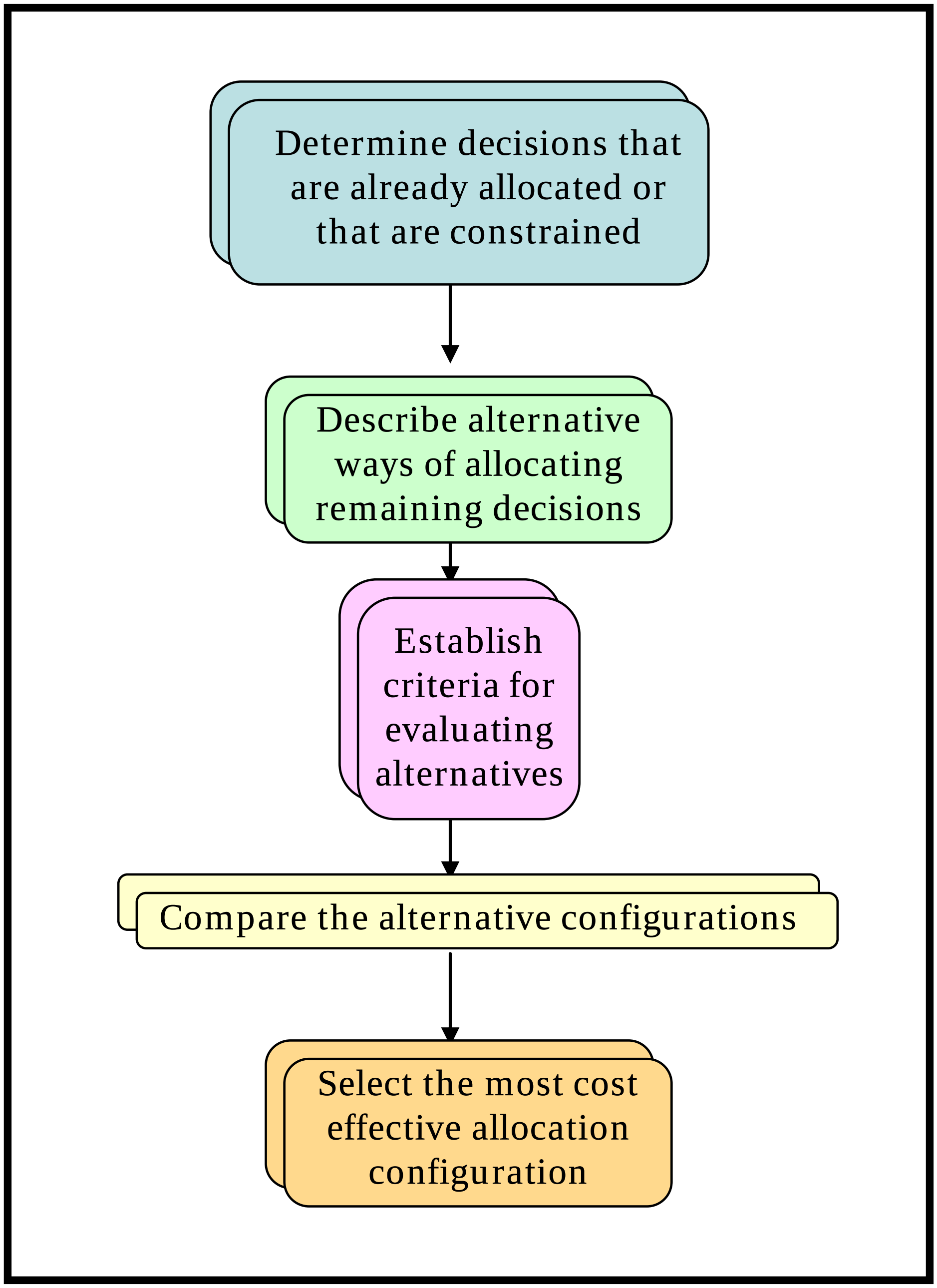


Figure 2-1: Five Stage Approach to Decision and Action Allocation

2.3 *Formal Approaches*

2.3.1 Fitts List

The first formal technique is based on what has come to be known as Fitts List [15]. It was created by P.M. Fitts in 1951 and has been widely referenced in literature since. His technique for determining which decisions should be carried out by machines versus which should be performed by humans is based on a simple dichotomous comparison of human and machine capabilities. He identified inherent strengths and weaknesses in both humans and machines. In many cases the strengths of one are the weaknesses of the other, so they compliment each other well. By exploiting the strengths and compensating for the weaknesses of both the human and computer, we are able to generate better solutions than either could produce alone.

Fitts' List, Table 2-1, was the original list of categories of man/machine capabilities, and it has been used as the baseline for many subsequent capability comparisons. Fitts' List is used as a guideline to produce an allocation of which decisions and actions are to be done by a human and which to are be done by machine, with each system decision being expressed in terms that allow the designer to associate it with one or more of the categories of man/machine capabilities contained in the list.

Humans appear to surpass present-day machines with respect to the following:
• Ability to detect small amounts of visual or acoustic energy • Ability to perceive patterns of light or sound • Ability to improvise and use flexible procedures • Ability to store very large amounts of information for long periods and to recall relevant facts at the appropriate time • Ability to reason inductively • Ability to exercise judgment
Present day machines appear to surpass humans with respect to the following:
• Ability to respond quickly to control signals, and to apply great force smoothly and precisely • Ability to perform repetitive, routine tasks • Ability to store information briefly and then to erase it completely • Ability to reason deductively, including computational ability • Ability to handle complex operations, i.e. to do many different things at once

Table 2-1: Fitts List

Machine capabilities have matured significantly in the past 55 years, making parts of Fitts' List obsolete. As a consequence, there have been many updates to the list over the years. For example, Table 2-2 is a list published 37 years later by the U.S. Department of Defense [32].

The advantage of using a capabilities list to determine who should make decisions is that it is simple and requires little training to use. In practice the list is a convenient framework for considering the allocation of decisions. It aids people unfamiliar with human factors to think systematically about the functions assigned to human operators. The list is a good first step or reference point when deciding which decisions should be allocated to a human and which should be allocated to a machine.

The disadvantages of relying solely on such a list are numerous. The main drawback is that the approach uses qualitative terms only. There is no quantitative metric to scale how much better a machine or human performs a certain action or what is actually defined as "a strength." In reality, when used alone, it is of limited help. It may also be difficult to relate the system actions or decisions to the limited list if the list is not comprehensive. In reality when deciding whom to allocate decisions to, there are numerous other trade-off factors to consider such as cost, affects on operators, support requirements, workload restrictions, etc. These factors are discussed in Section 2.4 as they are all *Evaluative Criteria* that should be used in finalizing the decision allocation.

HUMAN EXCELS IN	**MACHINES EXCEL IN**
Detection of certain forms of very low energy levels	Monitoring (both men and machines)
Sensitivity to an extremely wide variety of stimuli	Performing routine, repetitive, or very precise operations
Perceiving patterns and making generalizations about them	Responding very quickly to control signals
Ability to store large amounts of information for long periods, and recalling relevant facts at appropriate moments	Storing and recalling large amounts of information in short time periods
Ability to exercise judgment where events cannot be completely predicted	Performing complex and rapid computation with high accuracy
Improvising and adopting flexible procedures	Sensitivity to stimuli beyond the range of human sensitivity (infrared, radio, waves, etc.)
Ability to react to unexpected low-probability events	Doing many different things at one time
Applying originality in solving problems: i.e., alternative solutions	Exerting large amounts of force smoothly and precisely
Ability to profit from experience and alter course of action	Insensitivity to extraneous factors
Ability to perform fine manipulation, especially where misalignment appears unexpectedly	Ability to repeat operations very rapidly, continuously, and precisely the same way over a long period
Ability to continue to perform when overloaded	Operating in environments which are hostile to man or beyond human tolerance
Ability to reason inductively	Deductive processes

Table 2-2: More Recent Human-Machine Capability List

2.3.2 Price's 2D Scale

Another formal approach is the one developed by Price [25] that offers a slight alternative to the simple dichotomy of Fitts' List that is based on scaling human and machine capabilities. This two-dimensional model (Figure 2-2) rates human and machine capabilities from 'unsatisfactory' to 'excellent.' The resulting model identifies six different regions that correspond to different cases of human-machine capabilities' comparisons.

- In region 1, there is little difference in the relative capabilities of human and machine, and the decision allocation can be made on the basis of criteria other than relative performance.

- In region 2, human performance exceeds machine performance; the decision should be made by the human.

- In region 3, machine performance exceeds human performance; the decision should be made by the machine.

- In region 4, machine performance is so poor that the decision should be allocated to humans.

- In region 5, human performance is so poor that the decision should be allocated to machine.

- In region 6 the decision would be performed unacceptably by both human and machine, arguing for a different design approach.

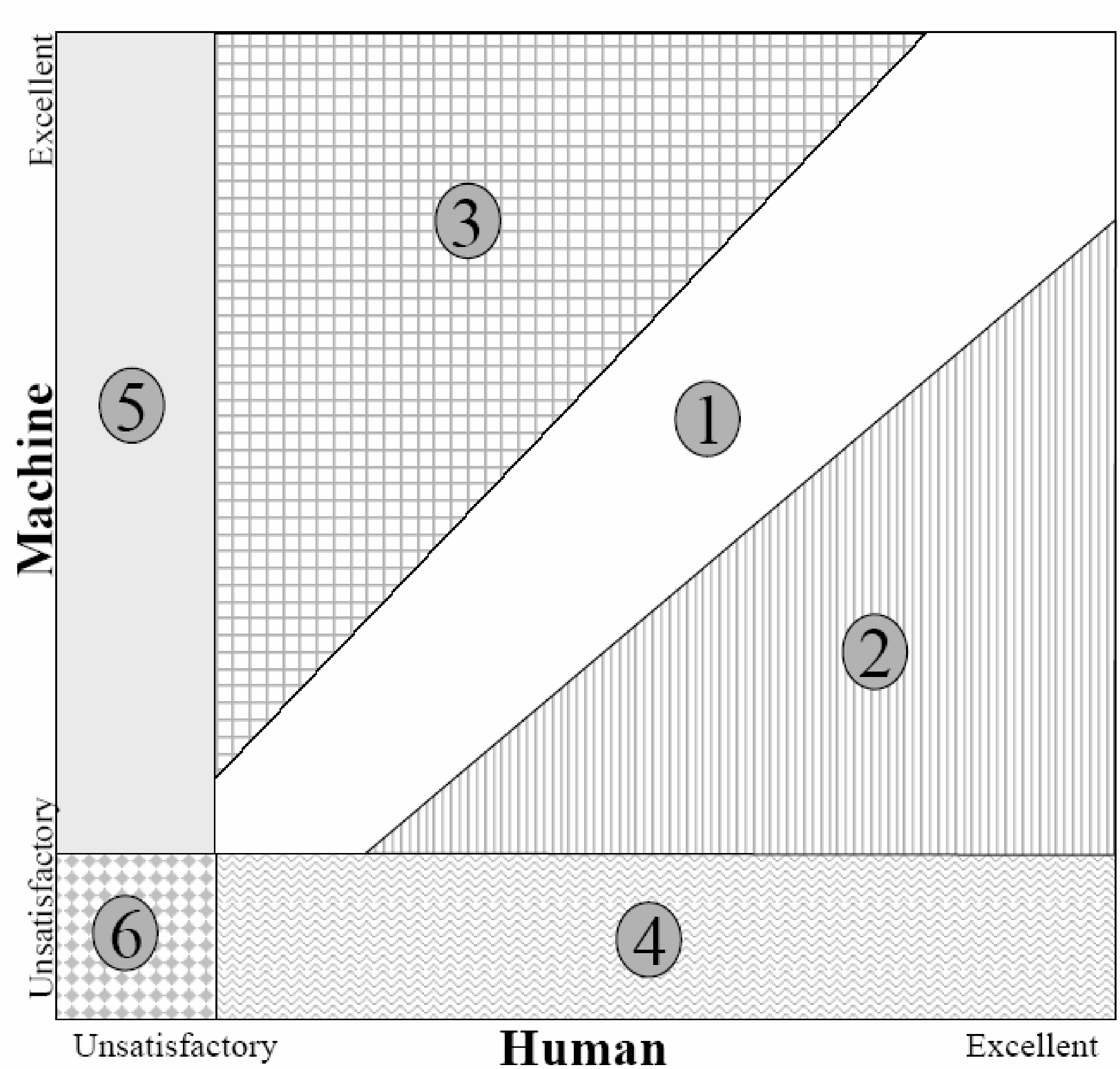


Figure 2-2: Criteria for Allocating Decisions to Human or Machine

Although this model supersedes the simple capabilities comparison listing suggested by Fitts, Price never mentions how one would go about deciding where on the subjective sliding scale to place the corresponding performance. Another obvious drawback of this approach is that what one person identifies as having ‘excellent’ human performance might be viewed by another as only having ‘very good’ performance. This could result in different regions (say region 2 vs. region 1) which would cause each separate user to come to a different conclusion regarding to whom to allocate the decision.

2.4 Sheridan & Verplank Autonomy Scale

Sheridan & Verplank enhanced previous approaches to decision allocation by incorporating the idea of collaboration between a human and machine to make decisions. They point out that a decision does not have to be made solely by a human or solely by a computer but that there are intermediate levels of automation that allow for cooperation between a human and computer. They have proposed that human-machine interaction can be characterized by a continuum of levels rather than as an all-or-none concept [29]. Under full manual control, a particular function is controlled by the human, with no machine control. At the other end of the spectrum corresponding to full machine control, the machine decides everything, including its own monitoring, ignoring any human input. Sheridan & Verplank's autonomy scale is presented in Figure 2-3.

For example, at Level 2 automation, the computer provides the human with several options but does not choose which decision will be made. At Level 4, the computer offers one potential alternative to the human but the ultimate authority on which decision to make lies with the human operator.

LEVELS OF AUTOMATION OF DECISION AND ACTION SELECTION

	Automation Level	Automation Description
	1	The computer offers no assistance: human must take all decision and actions.
LOW	2	The computer offers a complete set of decision/action alternatives, or
	3	narrows the selection down to a few, or
	4	suggests one alternative, and
	5	executes that suggestion if the human approves, or
	6	allows the human a restricted time to veto before automatic execution, or
	7	executes automatically, then necessarily informs humans, and
	8	informs the human only if asked, or
HIGH	9	informs the human only if it, the computer, decides to.
	10	The computer decides everything and acts autonomously, ignoring the human.

Figure 2-3: Sheridan and Verplank's 10 Level Autonomy Scale

2.5 Evaluative Criteria

Another idea proposed by Sheridan was that of using *Evaluative Criteria* when making the final determination of the level of automation. The purpose of the evaluative criteria is to account for intangible factors when determining how much control humans and machines should have over certain decisions. These criteria cover a variety of important factors ranging from human performance aspects to issues such as cost and levels of risk. Sheridan, Parasuraman, and Wickens broke down these criteria into the *Primary* and *Secondary Evaluative Criteria* [28] described in the following.

2.5.1 Primary Evaluative Criteria

- *Mental Workload* – A human operator can only handle a finite amount of work before reliability and production start to decline. One needs to measure mental workload in order to determine whether the induced workload exceeds the overall level of workload a controller can deal with effectively.

- *Situational Awareness* – Human situational awareness can either increase or decrease with increased automation. Increased automation can decrease situational awareness about the decisions that are being automated but can free the operator to provide more time to improve situational awareness by monitoring other actions or participating in other decisions throughout the system.

- *Complacency* – Complacency occurs when humans become over reliant on the machine. If the human over-trusts the automation, they might fail to realize the occasions when the automation fails. To prevent this, mechanisms should be established to provide the human with insight into decisions made by the automation.

- *Skill Degradation* – If a human user does not use a certain skill over a long period of time and is simply monitoring the computer's activities, there is a good chance the humans' skills will degrade. One has to question how sharp the user will be if there is an emergency or computer malfunction and the decision or action must suddenly revert from automated to manual. An example of skill degradation being taken into account occurs with airline pilots. In order to keep their skills sharp they are required to manually conduct a specified number of landings each month. If they were to use the autopilot to land every time, then eventually their aircraft landing skills would decrease.

2.5.2 Secondary Evaluative Criteria

- *Automation Reliability* – Increases in automation benefit a human by heightening their situational awareness to be used on other problems or decisions in the system. In systems where there are not other problems or decisions, these increases in automation can benefit by decreasing a human's mental workload. However, these benefits are unlikely to accrue if the computer or algorithm is unreliable. In unreliable systems, the mental workload actually increases for humans while their situational awareness decreases because time must be spent to determine if the results received were correct or not.

- *Costs of Decision/Action Outcome* – It is important to consider the costs that occur if the actions that the human or computer take are incorrect or inappropriate.

2.6 HMCDM Research Applied to Optimization

The research outlined above has a very general and broad scope. The views and ideas are aimed at any system that might have a human and machine component. The research focused solely on human-machine collaboration in solving optimization problems is much more limited. However, the research that has been conducted shows that HMCDM can be effective at producing improved optimization solutions [4] [27] [33]. The key issue in optimization problems is the same as it is in any more general system: determine the best division of labor between human and computer participants. Existing research in human-machine optimization has taken different approaches at establishing this division of labor.

Interactive evolution is an iterative approach wherein at each stage of the solution process the solutions are generated by the computer and the human selects which of these solutions will be used by the machine to generate new solutions in the next iteration [19]

[30] [31]. This process falls somewhere between Level 2 and Level 3 on Sheridan and Verplank's Autonomy Scale. Colgan *et al.* [14] developed a system wherein the human dynamically changes the parameters of the system used to evaluate candidate solutions. They describe a system for evaluating the choice of circuit design parameters where they allow the user to visualize the interrelationships and sensitivities of the various parameters. The focus is on helping the user identify the parameters that are most important to study, rather than on choosing specific parameter values. Other approaches permit users to manually modify the computer-generated solutions [33] [12]. They place little or no restrictions on the human-initiated modifications, and employ heuristics to resolve constraint violations that may be introduced by the user.

Research on human-machine collaboration in optimization has also been conducted at the Mitsubishi Electric Research Laboratory. Their studies suggest that humans are effective at guiding and focusing searches, managing machine computational effort, and visually identifying promising areas of a solution search space [4] [27].

2.7 Additions and Enhancements to Previous HMCDM Research

2.7.1 Current Capability Strength Lists

Many of the capabilities identified as human and machine strengths by Fitts and the DoD are still valid. However, there are capabilities that were never mentioned in these earlier lists that seem applicable today. For example, humans are particularly strong in areas such as communicating complex ideas, symbolic reasoning, conceptualization, learning from experience, and intuition. Humans are able to store and adapt experience and quickly grasp the overall picture of complex situations. Their ability to recognize patterns is applicable not only to visual stimuli but also to abstract concepts and intuitive notions.

Although the biological basis of our cognitive abilities is massively parallel, our conscious reasoning capabilities are essentially sequential [11]. Therefore, human decision makers are easily overwhelmed by large volumes of information and very complex decision scenarios where each decision may have many unobvious interactions with other decisions or actions. It is difficult for a human to analyze more than three or four variables at any one time, especially when these variables interact (combinatorial problems). Under these circumstances, humans tend to switch from an analysis mode to an intuitive mode in which they rely almost completely on their ability to develop situation awareness and make decisions through abstraction and conceptualization. While this is a noteworthy strength of humans it is also potentially a great weakness. At this intuitive level, instead of working objectively toward an optimal decision, humans are vulnerable to emotional influences that are an intrinsic part of human nature [11].

The tendency to rely on emotions can make the human operator somewhat unpredictable and resistant to dynamically changing situations. Confidence in one's ability to deal with complex and critical situations is based to a large extent on past experience with similar problem situations. Therefore, if the situation is continually changing, humans are less likely to be able to rely on past experiences and as a result feel less confident in being able to successfully deal with the changed situation.

Computer capabilities are strongest in the areas of data management, speed and accuracy. Machines also excel in parallelism; meaning that a computer can conduct multiple functions and calculations at once whereas humans have the problem of losing focus or becoming confused as to which task they are working on. Computers also have an enormous capacity for storing data. While a human is prone to making minor mistakes in arithmetic and reading, the computer is always accurate. For example, a slight diversion may be sufficient to disrupt a human's attention to the degree that causes the incorrect adding or subtracting of two numbers. However, human's make up for this weakness by being able to notice when large errors occur due to the ability to use common sense. Unfortunately, the computer cannot distinguish between a minor mistake and a major error.

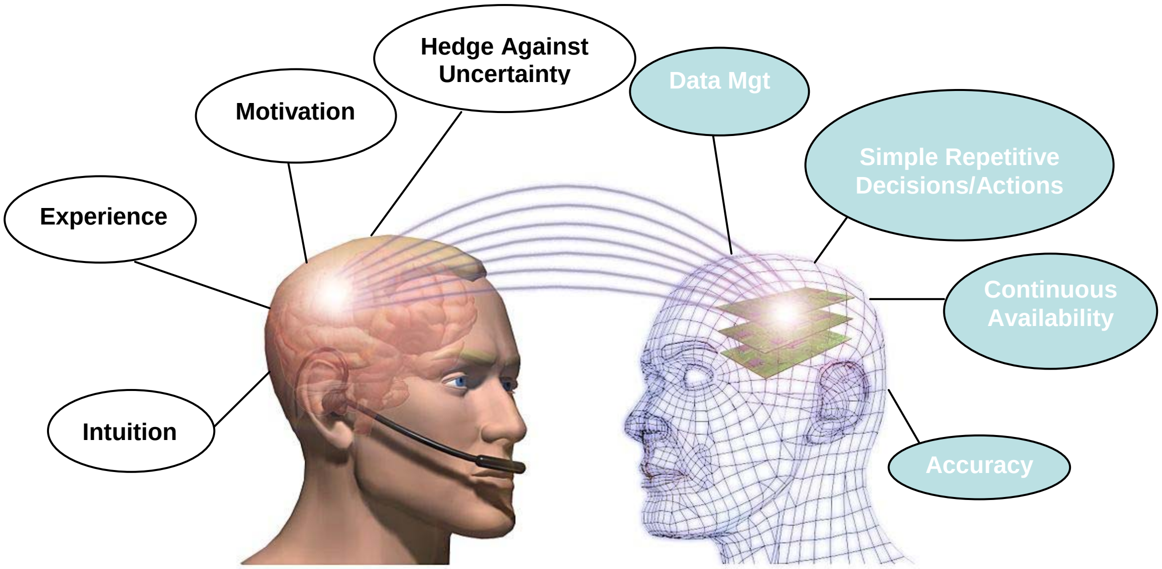


Figure 2-4: Human and Machine Capability Strengths

Sections 2.7.1.1 and 2.7.1.2 provide listings of current human and machine capabilities. These listings are not meant to be inclusive of all human and machine capability strengths. Rather they are provided to highlight the relevant capabilities that were taken into consideration in the problem application presented later in the thesis.

2.7.1.1 Human Capability Strengths

Flexible/Adaptable – Humans are much more flexible and adaptable to situations that arise in real life. Computers will only do what they are programmed to do. Humans are able to adjust readily to changing conditions.

Creativity – Humans have the ability to think outside the box and display originality and imagination. They are also able to apply off-topic knowledge that may be useful to the situation.

Visual Perception – Humans are excellent at visual perception. In fact, human reasoning and learning abilities stem, in part, from our ability to visually perceive. On the other hand, while objects can be fairly easily represented in the computer as visual images and data relationships, the computer has a great deal of difficulty in understanding their real world meaning.

Emotion – This allows a decision maker to consider abstract concepts that might not be easily modeled by a computer instead of relying solely on a computer's objective output.

Learning From Experience – A human can analyze a situation or scenario and rely upon their previous experience to help them in two ways. They can either quickly narrow the "search space" of options by discarding options that they know will not work in a certain situation or they can make a concrete decision by knowing that a particular action will work based on prior experience. Although there has been significant progress in the AI community with respect to machine learning, humans still clearly are superior on this dimension.

Complex Communication – A computer is limited in its ability to communicate with a human via sound through the speakers and printouts or visualizations on the screen. Inter-Human communication on the other hand is much more complex involving such things as tone of voice, hand and face gestures, and mannerisms just to name a few.

Conceptualization – Humans have the ability to invent or contrive an idea and formulate it mentally. This is the mental process whereby fuzzy and imprecise notions are made more specific and precise. Even with recent advances in artificial intelligence, computers are not able to compete with a human in this area.

Symbolic or Spatial Reasoning – This strength is related to strengths of visual perception in that humans are able to relate to and understand information and scenarios using visual stimuli only. Humans have the ability to manipulate abstract symbols mentally and use them to make judgments and decisions that are logically valid.

Intuition – The fact that humans have seen many different situations throughout their lifetimes and can remember experiences gives them instinct and allows them to rely on this intuition when making decisions. This instinctive knowledge can reduce the amount of time needed to make a decision. Without the benefit of intuition, computers are forced to proceed through a possibly more time consuming rational process.

Pattern Recognition – This capability refers to the classification or description of observations. For the purposes of our research, we are specifically interested in a human's ability to recognize patterns with respect to cluster analysis. Recent work in the AI community has increased the machine's capability to recognize patterns, but the human still has the edge in this area. A human can also recognize patterns of behavior and extrapolate to predict future behavior.

Hedging Against Uncertainty – A prerequisite for being able to hedge against uncertainty is the ability to anticipate "possible" future states. People have this ability to guess where things can go wrong and hedging against these possible problems. Humans have a store of what is often referred to by the AI community as "common sense knowledge" [21], that they bring to bear in assessing what might go awry in a given situation. It's difficult to model in the computer all of the things that might go wrong – and thus, computer methods cannot predict what hasn't been modeled. This is particularly true to the specific air operations application presented in this thesis. If we tried to code up an exhaustive list of all possible things that could go wrong or 'possible states of the world,' and then solve a stochastic program on these states of the world, it would be computationally intractable. On the other hand, if a human looks at the scenario and realizes where the problems might occur with a high degree of probability and feeds this information to the computer, the computer solution might be much better and would certainly be obtained much faster.

Narrowing Search Space – Experiments done by Mitsubishi Electric Research Labs (MERL) [4] [27] have shown that a human can effectively narrow the space to search for optimal solutions. The experiments were run with humans in the loop of a capacitated vehicle routing with time windows problem.

Management of Computational Effort –MERL's experiments [4] [27] also showed that a human operator is very effective in managing the computational effort expended on problems. They showed that humans are able to accurately guess the point of diminishing returns in the search for a solution.

Strategic Assessment – Compared to a computer, a human is able to bring a broader range of strategies to the table. It is impossible for a computer to employ problem solving strategies beyond that which it is programmed to consider. A human has the ability to think critically about the situation and consider more strategies.

Understanding the "Big Picture" – Many of a human's other strengths combine into making a human good at understanding the big picture. The *big picture* refers to understanding the impact of the solution on the world outside of the "system" for which that solution has been developed. This is something a computer cannot do well.

2.7.1.2 Computer Capability Strengths

Displaying Information – This could be a geographical representation of information or any other visual representation of data or information. For example, when applied to air operations, a computer can generate a graphical representation of where all of our targets are located on a map and a visual list of the targets to help facilitate clustering. A computer is better, more flexible and faster at generating a display of this information than is a human.

Data Management – Includes storing and retrieving data - the only limiting factor in the amount of data to be stored in a computer is its' own internal hard drive capacity.

Simple Repetitive Decisions – Once a computer is programmed to perform a certain action or calculation, it can do so whenever asked. This makes computers very effective at performing simple repetitive decisions. In contrast, there is a chance that humans may make small errors or not perform the action in the exact same manner each time.

Performing Calculations – Computers are better at doing mathematical calculations. They are both more accurate and faster than a human in this area especially when performing long or complex calculations.

Combinatorial Problems – These are problems with a large number of variables. Due to the extremely large nature of combinatorial problems, a computer is much better at solving these types of problems. It is very difficult for a human to obtain a solid grasp on the problem because of the sheer number of possible solutions.

Continuous Availability – Computers are an untiring resource. They can be utilized 24 hours a day, 7 days a week, 365 days a year.

Fast Computational Parallel Reasoning – Computers are able to simultaneously conduct numerous operations at the same time. For example, a computer is able to perform numerous calculations simultaneously whereas most humans must perform them sequentially. It is more difficult for a human to "multi-task."

Speed – Computers are much faster than humans at many different tasks. High levels of computer involvement could be necessary in time-critical situations in which there might not be adequate time for a human to respond and take appropriate action.

Accuracy – As mentioned earlier, humans are prone to making minor mistakes in arithmetic, whereas the computer is always accurate.

Predictability – A computer performs what a human has programmed it to do. In addition to being a drawback as mentioned earlier, this also provides a benefit in the form of predictability. This predictability stems from the fact that computers are built on a simple '0' and '1' system. There is no degree of vagueness here, '0' and '1' are precise digital entities and very different from the massively parallel and largely unpredictable interactions of neurons and synapses that drive human behavior.

Low Cost – One of the main drivers behind the desire to raise levels of automation in systems today is that it can be cost-effective. There may be a high initial cost, but automating tasks and decisions can be a good long term investment.

Risky Situations – Computers or machines can be an effective means to replace humans in risky situations. A military related example is the recent push to increase the use of unmanned aerial vehicles in place of piloted fighter aircraft to accomplish dangerous missions, risking the possible loss of a machine rather than a human life.

2.7.2 Proposed Additional Evaluative Criteria

In addition to Sheridan, Parasuraman, and Wicken's *Evaluative Criteria* [28], we propose below additional factors that should be considered in deciding who should be making what decisions.

- Human Tendency for Boredom – This criterion refers mainly to when a human is placed strictly in a monitoring role. If a human does not have an active role in the decision making process, this could result in a reduction in the complexity of human interaction with the system, to the point of boredom. This boredom could result in tasks not being performed reliably over long periods of time.

- Trust – It is important that the human operator trust the decisions or solutions that a computer generates. This factor is usually tied into the reliability of the machine or computer and the correctness of the associated software making the decisions. One thing to consider is that users are more likely to understand and accept a solution that they helped create, as compared to one presented to them with no insight as to how the solution was reached (example of under-reliance on the machine). It is also important to be wary of the opposite case in which humans develop too much trust for a system and become complacent (over-reliance on the machine). Complacency can be combated by ensuring the human must control a portion of the critical decisions in the system.

- Skill Set Requirement – The skill set required of the human operator can be a function of the level of automation at which decisions are made. This is especially true if the system has been in place and it is then decided to change who is making the decisions (between the computer and the human). Some formerly required skills may now be obsolete while the need for new skills arises.

- Human Team Dynamics – Often decisions are made by teams of humans. More or less automation for certain decisions and actions might cause changes in the team structure and composition, redefine team roles, or alter interaction and communication patterns. This is particularly of interest in our research due to the outcome of our experiment. The results suggest that teams of humans interacting with a computer might add more benefit than lone human users. However, if this is to be tested in the future, it should be understood that there are certain dynamics within a team that could affect the hypothesized benefit.

- Human Operator Prior Experience – Prior experience of the operator is likely to be a factor affecting the successful implementation of automation levels. If the humans have little to no experience, they must be taught from the ground up. On the other hand, if the human operators have significant previous experience and the majority of it comes in systems that have mostly been under manual control, they might have difficulties adapting to a system in which many decisions are highly automated. The same is true for humans coming from a system in which decisions are highly automated to one under full manual control. It will be harder to re-train these older workers who are already set in the ways of how procedures were done in the past. In these cases, it might be better to find humans with little to no prior experience and train them accordingly. This was taken into consideration before running the experiments for this thesis. It was decided that all participants would have no experience using a similar system.

- Recovery From System Failure – Automation can be designed to reduce or eliminate certain human errors. However, higher levels of automation may also lead to new classes of human errors related to reduced situational awareness. The user needs to understand what is going on throughout the decision process even if the majority of the decisions are made by the machine so that in the event of failure of the automation support, they will understand how to ensure a safe recovery.

- Decision Interactions – The interdependencies among decisions (how certain decisions affect each other) must be analyzed before making a final choice of the levels of automation. The choice on which level of automation to place a decision is not as trivial as analyzing that decision independently. It is likely that numerous decisions throughout a system affect each other at least in an indirect manner.

- <u>Decisions Involving High Risk</u> – The risk associated with a decision outcome can be defined as the cost of an error multiplied by the probability of that error. Decisions with little inherent risk (low cost and/or low probability of error) are strong candidates for high-levels of automation. According to Parasuraman, Sheridan, and Wickens, "If human operators had to be continually involved in making each of these relatively simple decisions, they could be so overloaded as to prevent them from carrying out other more important functions" [28]. On the other hand, decisions with higher levels of risk, such as those considered in our research and the application in this thesis, need to be studied to determine the appropriate level of human involvement

- <u>Responsibility</u> – The level of responsibility placed on the operator for the consequences of a decision outcome will have a significant impact on the degree to which the operator accepts the decisions made by automation. Thus, it will be important to design mechanisms that give the operator insight into the basis for the decisions made by automation.

2.8 Proposed Methodology for Determining Level of Human-Machine Collaboration

We have combined ideas from previous researchers with our own thoughts in formulating a methodology for determining the appropriate levels of human machine collaboration. The methodology for allocating decisions that we propose is a "balanced approach" in that it first accounts for decisions that need to be placed with either the human or machine for political or managerial reasons, and performs a formal analysis on the remaining decisions to determine their allocation. In addition, our methodology overcomes limitations of previous approaches in that we have developed a quantitative tool that guides the allocation determination. We refer to this tool as the *Human Machine*

Collaboration Worksheet. The steps of our proposed methodology are outlined below in Table 2-3 and Figure 2-7 and are described in detail in the following subsections.

1. Identify the decisions to be made throughout the system.
2. Identify and **Remove** decisions that are constrained to be performed in a pre-specified manner (e.g. for political, managerial reasons, etc).
3. Create listing of human-machine capability strengths.
4. Assign Pairwise Comparison Weights to the Strengths.
5. Use the weights to score each decision as to how well these strengths apply for a human performing a certain action/decision and for the computer performing the same action/decision.
6. Based on these two weighted scores, pick a level of autonomy from Sheridan's Autonomy Scale.
7. Scrutinize the decisions for the levels of autonomy based on "primary" and "secondary" evaluative criteria to be considered.
8. Finalize the level of automation.

Table 2-3: Proposed Algorithm for Determining Human-Machine Involvement in System Decisions & Actions

2.8.1 Step 1 – Identify Decisions and Actions

The first step in our methodology is to create an exhaustive list of the decisions that will be made in order to have a full understanding of the system. This list is created from the knowledge of what the system is intended to produce. By understanding the purpose of the system, it is possible to work backwards to determine which decisions will need to be made in order to reach the desired outcome.

2.8.2 Step 2 – Reduce List to Unconstrained Decisions/Actions

Step 2 is derived from Meister's [22] view that there are certain decisions which are constrained to be performed a particular way. These decisions could be constrained for any number of political or managerial reasons. The constrained decisions are removed from the list of decisions created in Step 1, and the formal analysis is conducted on only those decisions that have not been removed.

2.8.3 Step 3 - Create Human-Machine Capability Strengths List

The third step in our methodology is to create a list of human and machine capability strengths. The list provides a good initial understanding of the inherent strengths and weaknesses of humans and machines that allows us to exploit the strengths and compensate for the weaknesses of both the human and computer in order to generate better solutions than either could produce alone. We have provided such a capabilities listing in Section 2.7.1. However, this listing is not meant to be exhaustive; it highlights the capabilities that are relevant to our research in the area of optimization and our specific application which is introduced in Chapter 3.

2.8.4 Step 4 - Assign Pairwise Comparison Weights to the Strengths

Next, instead of allocating decisions based solely on these strengths as Fitts and Price did, we obtain an assessment of how these "strengths" compare to each other. For example, the human's strength of *experience* may be more important than the strength of *creativity*. Numerical weightings are given to each respective strength based on a pairwise comparison with other strengths. These weightings can be derived using subjective (subject matter experts) or objective (cost) measures or any combination of the two. It is important to note that there are not a broad generalized set of weights. The weights depend on the amount and type of subjective and objective measures used.

One approach to conducting the comparison is through the use of a *Paired Comparison Chart* which is shown in Figure 2-5. In this approach, each strength is compared with each other in a matrix, and given a value of '1' if the strength in the column is more important than the strength listed in the corresponding row, '0' if the strength in the column is less important, and '0.5' if the strengths are of equal importance. Again, this numerical determination depends on the amount and type of subjective and objective measured used. The column values in the *Paired Comparison Chart* are summed for each strength and the weighting factor for each is then obtained from the normalized summed value for each strength. This procedure is to be performed separately to obtain the respective weights for both human and computer strengths.

For instance, Figure 2-5 provides an example of a *Human Capability Paired Comparison Chart.* In this example, the human strength of *learning from experience* is considered to be more important than *intuition*, *recognizing patterns, hedging against uncertainty*, and *creativity*. This is represented by the four values of “1” in the *learning from experience* column. *Recognizing patterns* and *intuition* are considered of equal importance, shown in the chart by a value of “0.5” in the corresponding boxes.

For instance, in the example provided in Figure 2-6, suppose the maximum weight score possible is ‘5’ (as is the case in the example provided in Figure 2-6). Summing each column gives us a score of ‘4’ for the *learning from experience* strength. In this example, each of the strengths is compared against four unique strengths, therefore each summed strength value will be divided by the number four and multiplied by the maximum weight value of five. Thus, the *learning from experience* strength is calculated in the following manner:

$$strength\ weight = \frac{\sum_i strength_i}{number\ of\ unique\ strengths} * maximum\ weight\ value \qquad (2.1)$$

$$learning\ from\ experience\ weight\ = \frac{1+1+1+1}{4} * 5 = 5$$

This process is repeated until a weight has been assigned for each strength.

Paired Comparison Technique for Determining Strength Weights

	Learning from experience	Inductive Reasoning (Intuition)	Recognizing Patterns	Hedging against uncertainty	Creativity	etc…
Learning from experience		0	0	0	0	…
Inductive reasoning (Intuition)	1		0.5	0	0	…
Recognizing patterns	1	0.5		0	0	…
Hedging against uncertainty	1	1	1		0	…
Creativity	1	1	1	1		…
etc…	…	…	…	…	…	

Figure 2-5: Human Capability Paired Comparison Chart. Used for Determining the Weighting Factors on Strengths Used in *Human Machine Collaboration Worksheet*

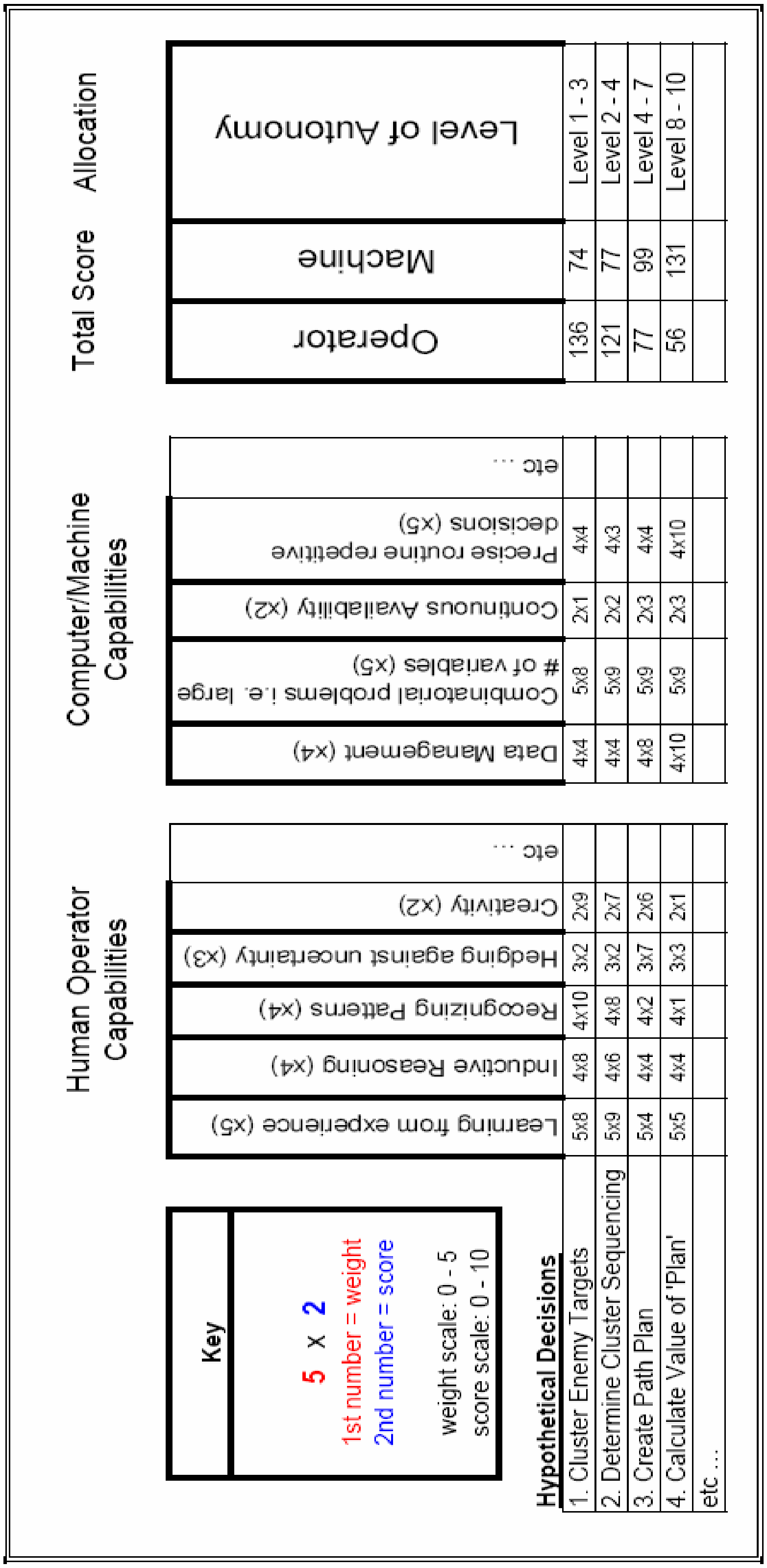
2.8.5 Step 5 – Score Strengths based on their Impact on each Decision

Step 5 introduces a quantitative measure of the contribution of each strength to the decisions or actions in the list developed in Step 2. For example, suppose the strength in question is *recognizing patterns* and that two of the decisions or actions are: 1.) create clusters of objects using the geographic layout of the objects and 2.) perform a numerical calculation. It should be obvious that *recognizing patterns* benefits each of the two decisions differently. In the case of clustering the objects, *recognizing patterns* is relevant and would help in carrying out the action. However, *recognizing patterns* does not provide much benefit in performing a numerical calculation. Therefore, the human strength of *recognizing patterns* will receive a large score for the clustering decision and a small score for the calculation action. These scores will range from 0 to 10 (see Figure

2-6). The values assigned to the scores are subjective – it is their relative values that are important.

The decision impact score for each strength is multiplied by the pairwise comparison weighting factors determined in Step 4. The product of these two values is the *Weighted Decision-Strength Score* (WDSS). Every decision listed in Step 2 will have a WDSS for each human and computer strength, and these WDSS's are then added together to provide an overall assessment of the relevance of human and computer strengths for each identified decision for the application of interest. Thus, the result is a total score for the human and a total score for the computer for each decision or action to be made in the system.

A useful visual organization of the overall assessment of the relative strengths that includes the pairwise weightings, impact scores, and WDDS's is the *Human Machine Collaboration Worksheet* depicted in Figure 2-6. The decisions and actions identified in Step 2 are listed in rows on the left hand side of the worksheet. Two sets of columns are listed across the top of the sheet; the first corresponding to human operator capability strengths and the second to machine strengths derived from the list created in Step 3. The pairwise comparison weights obtained in Step 4 are listed after the name of each strength and as the first number in each box of their respective column. The second number in each box is the impact score that reflects the degree to which each strength (listed in columns) contributes to the decision or action (listed in rows). This score was determined in Step 5. The far right of the worksheet contains the total weighted scores (WDSS's) for the human and machine for each decision or action. Also listed is the range of levels of autonomy in which to place that particular decision. This is discussed below in Step 6.



Hypothetical Decisions	Human Operator Capabilities: Learning from experience (x5)	Inductive Reasoning (x4)	Recognizing Patterns (x4)	Hedging against uncertainty (x3)	Creativity (x2)	etc ...	Computer/Machine Capabilities: Data Management (x4)	Combinatorial problems i.e. large # of variables (x5)	Continuous Availability (x2)	Precise routine repetitive decisions (x5)	etc ...	Total Score: Operator	Machine	Allocation: Level of Autonomy
1. Cluster Enemy Targets	5x8	4x8	4x10	3x2	2x9		4x4	5x8	2x1	4x4		136	74	Level 1 - 3
2. Determine Cluster Sequencing	5x9	4x6	4x8	3x2	2x7		4x4	5x9	2x2	4x3		121	77	Level 2 - 4
3. Create Path Plan	5x4	4x4	4x2	3x7	2x6		4x8	5x9	2x3	4x4		77	99	Level 4 - 7
4. Calculate Value of 'Plan'	5x5	4x4	4x1	3x3	2x1		4x10	5x9	2x3	4x10		56	131	Level 8 - 10
etc ...														

Figure 2-6: Example of *Human Machine Collaboration Worksheet*

2.8.6 Step 6 – Choose Level of Autonomy

Based on the human and machine total weighted scores for each decision or action (WDSS's), we determine a range within Sheridan and Verplank's Autonomy Scale to assign to the decision/action allocation. This is done subjectively as there is not a set rule which links a particular score to an exact level of automation. However, those decisions for which the human WDDS scores are much higher than machine WDDS scores should be placed somewhere at the lower levels of automation ("Cluster Enemy Targets" decision in Figure 2-6). These lower levels of automation correspond to decisions being carried out solely by humans or mainly by humans with small augmentation or collaboration from a machine. Similarly, decisions whose machine WDDS scores are much higher than human WDDS scores should be placed at higher levels of automation ("Calculate Value of Plan" decision in Figure 2-6). Higher levels of automation reflect decisions being carried out primarily by the machine with little to no involvement from a human. Those decisions with roughly equivalent human and machine WDDS scores should be placed somewhere in the middle of Sheridan and Verplank's Autonomy Scale. Refer to Figure 2-3 for the description of each level of automation.

2.8.7 Step 7 – Scrutinize Level of Autonomy with Evaluative Criteria

After performing the Step 6 analysis of the *Human Machine Collaboration Worksheet*, we are left with a range of Sheridan and Verplank's automation levels for which to place each decision or action. It is not realistic to strictly assign decisions to a human or a computer based solely on the score they receive in the allocation worksheet. In order to narrow this range to a single level of automation, we use Sheridan's, as well as our own *Evaluative Criteria*, described in Sections 2.5 and 2.7.2, respectively. Again, the purpose of these criteria is to take into account intangible factors that might not be easily quantifiable. Figure 2-6 shows a *Human Machine Collaboration Worksheet* which provides a small example applied to the Mixed-Initiative Control of Autonoma-teams (MICA) problem which is used as the application test bed later in the thesis. Figure 2-7 recaps the proposed methodology for determining the level of human-machine collaboration.

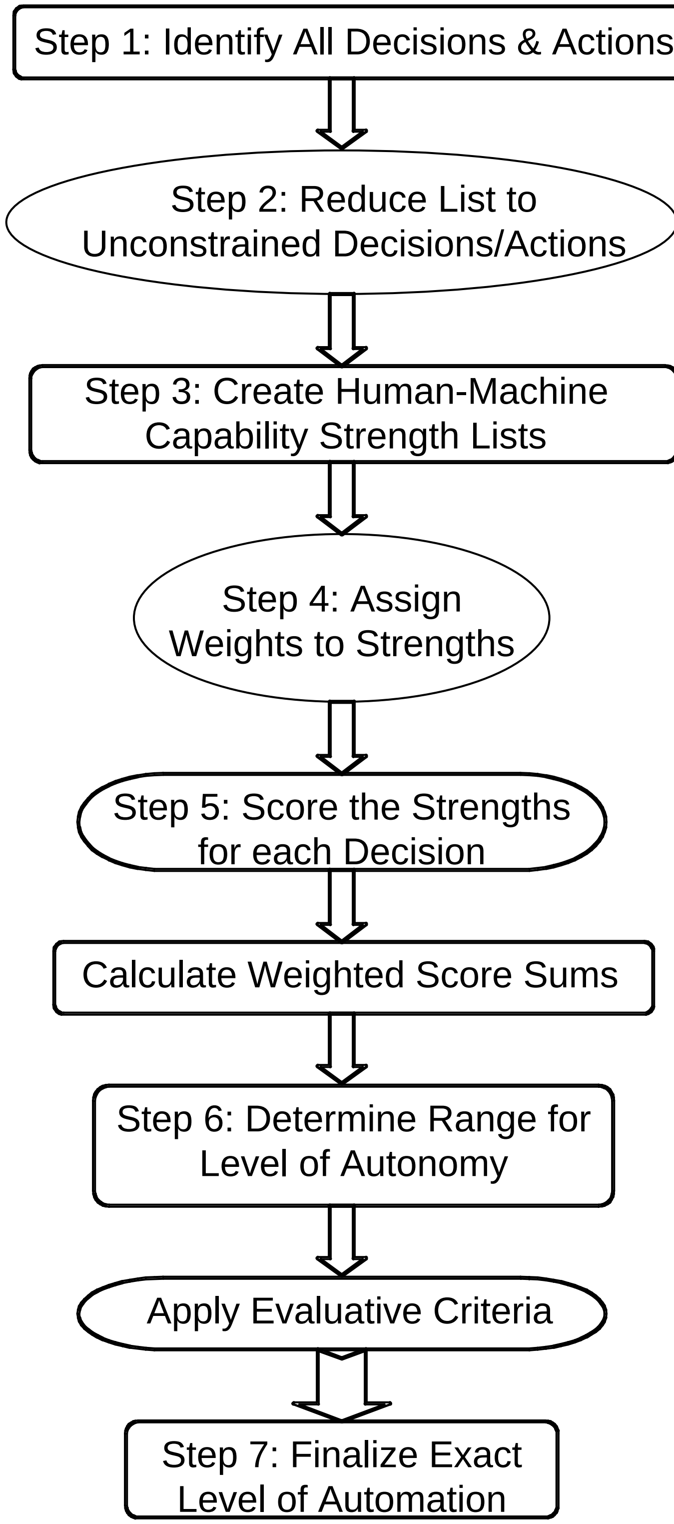


Figure 2-7: Proposed Algorithm for Determining Human-Machine Involvement in System Decisions & Actions

Chapter 3

Mixed-Initiative Control of Automa-teams

The purpose of this chapter is to present the MICA resource allocation and planning system for coordinating actions among unmanned aerial vehicles (UAVs). We describe the motivation for the MICA system and conclude the chapter with an analysis of the benefits HMCDM can provide to specific subproblems within the MICA system.

Optimization problems of the class addressed in the MICA program are NP-complete [2] [9]. Extremely long execution times are required in order to solve such problems to optimality. This is unacceptable for problems wherein timely solutions are required to accommodate changes in the environment. An approach that has proved to be effective in problems of this class is referred to as *composite variable formulations* [5]. It addresses the intractability issues by combining many of the decision variables in the original problem into composite variables which each represent a collection of these variables. In MICA, the decision variables for the original problem are: for every time interval, where should each Unmanned Aerial Vehicle (UAV) be and what activity should it be performing [2]. The composite variable formulation for MICA encompasses all decisions required for the complete mission for a team of UAVs. These composites, which can be viewed as "plan fragments" of the overall MICA plan for all vehicles and all teams, are also referred to as "options."

The advantage of the composite variable formulation is that it can be easier to solve than the original problem. There are however two main challenges to using such an approach: defining the right mapping from the original variables to composite variables and selecting which composite variables to generate [1]. In the case of MICA, there are an exponential number of possible missions for each possible team of vehicles when considering every possible set of targets. The clustering and resource allocation approach described throughout the chapter has been developed to select a good set of team options to consider in the final solution.

3.1 Problem Being Solved in MICA

MICA is initialized with a list of assets (Blue entities), resources associated with those assets (weapons and sensors), enemy threats and targets (Red entities), and Commander's Intent (described below). Based on this information, the goal is to *select, sequence,* and *schedule* sensing and strike activities for the available aircraft resources to prosecute enemy targets in an effort to maximize the total expected *Value* minus *Cost* (described below). The solution developed for the MICA problem is a closed-loop, dynamic planning and execution system for selecting courses of action (COA) for UAV mission planning.

- *Value* is computed as a function of commander's intent. Commander's Intent is described as a function of three intent matrices; the ***Awareness Intent Matrix (AIM)***, the ***Kill Intent Matrix (KIM)***, and the ***Damage Intent Matrix (DIM)*** [2]. Each of these is discussed further in Section 3.2.1.1.

- *Cost* is computed as a function of aircraft loss, aircraft detection, and cost of resources used.

- Both *Value* and *Cost* are expected values with respect to the battlespace state, sensor performance, weapon performance, and enemy (Red) air defense system performance [2].

More formally, the COAs for MICA are determined by solving a large-scale binary program. The binary decision variables are composite variables that represent a combination of which aircraft are responsible for what enemy targets in each time period. Using these composite variables, the problem is formulated as the following integer program [1]:

$$
\begin{aligned}
\max \quad & v^T x \\
\textit{subject to} \quad & Ax = b \\
& x \in \{0,1\}^n
\end{aligned}
\tag{3.1}
$$

where $\boldsymbol{x}$ is a $\boldsymbol{n}$-dimensional binary decision and $\boldsymbol{x_j}$ is equal to 1 if the j^{th} *option* is selected, and 0 otherwise. We consider $\boldsymbol{x_j}$ a composite variable since $\boldsymbol{x_j} = 1$ means that *option j* and all the decisions that compose *option j* have been selected. An example of an option would be two aircraft of commodity type #1 performing actions (strike and/or sense) on targets within cluster #4 (which could include multiple enemy targets) flying a specific flight pattern in time period #2. More detail on the building of options is presented throughout this chapter. The variable $\boldsymbol{v_j}$ represents the (value - cost) of *option j*, and factors in the value of the targets in the plan, the risk of attrition, and fuel usage. The constraints $\boldsymbol{Ax} = \boldsymbol{b}$ can be broken into aircraft constraints (one for each aircraft commodity type in each time period) and target constraints (one for each target). These constraints ensure that plans are not created with more aircraft flying missions than are available in each time period and that each enemy target is assigned to exactly one cluster. More explicitly, $\boldsymbol{A}$ is composed of $\boldsymbol{n}$ options where each column $\boldsymbol{A}_\mathbf{j}$ of $\boldsymbol{A}$ represents an option. Figure 3-1 provides a further break down of $\boldsymbol{A}_\mathbf{j}$. Each parameter in the figure takes on the value 0 or 1. If the value equals 1, then the designated action is carried out. For example if the parameter $\boldsymbol{a^{11}} = 1$, then aircraft commodity 1 is used in period 1. In addition, $\boldsymbol{A}$ is the total number of aircraft commodities, $\boldsymbol{T}$ is the number of time periods, $\boldsymbol{H}$ is the total number of possible targets to hit, $\boldsymbol{I}$ is the total number of possible targets to optically image (sense), and $\boldsymbol{G}$ is the total number of “grid cells.” The combination of areas to search (grid cells) and enemy targets are known as Points of Interest (POIs). In order to solve this problem, the large-scale optimization problem is broken down into smaller, more tractable subproblems. The coordinating and solving of these subproblems creates COAs as described in the following sections.

$$
\mathbf{A}_j = \begin{bmatrix}
a^{11} = \text{aircraft commodity 1 used in period 1} \\
\vdots \\
a^{kl} = \text{aircraft commodity } k \text{ used in period } l \\
\vdots \\
a^{AT} = \text{aircraft commodity } A \text{ used in period } T \\
h^{1} = \text{target 1 hit} \\
\vdots \\
h^{f} = \text{target } f \text{ hit} \\
\vdots \\
h^{H} = \text{target } H \text{ hit} \\
i^{1} = \text{target 1 optically imaged} \\
\vdots \\
i^{d} = \text{target } d \text{ optically imaged} \\
\vdots \\
i^{I} = \text{target } I \text{ optically imaged} \\
g^{1} = \text{grid cell 1 searched} \\
\vdots \\
g^{c} = \text{grid cell } c \text{ searched} \\
\vdots \\
g^{G} = \text{grid cell } G \text{ searched}
\end{bmatrix}_j
$$

Figure 3-1: MICA problem

3.2 The MICA Three-Tiered Planning Hierarchy

The MICA system is decomposed into a three-tiered planning hierarchy (Figure 3-2 [2] and Figure 3-3 [2]). This hierarchical structure breaks down the large-scale optimization problem into smaller, more easily solvable subproblems in an effort to balance optimality and tractability. Each of the three tiers or levels is responsible for a separate subproblem or aspect of the overall larger planning optimization problem. Starting at the top, the *Team Composition and Task Allocation* (TCTA) level is given all available information about the problem. This includes *commander's intent*, information about all of the available friendly (Blue) resources, and information about the enemy targets. Given this information, TCTA groups enemy targets into clusters and assigns teams of aircraft resources to these clusters. We refer to these aircraft resource – cluster pairs as"options." These resource–cluster pairs are sent to the next level of the planning hierarchy which is termed *Team Dynamics and Tactics* (TDT). The TDT level determines the sequencing of the enemy targets contained within each separate cluster. This level also establishes the best set of sensing, strike and self-protection activities to be performed in prosecuting each target in order to maximize the value achieved while minimizing cost. The lowest level of the planning hierarchy is called *Cooperative Team and Path Planning* (CTPP). This level adds further details to the plans developed for each option at the TDT level, including sensor pointing, weapon selection, weapon release time and location, cooperative team self-protection jamming and the exact routes of each vehicle in the team. The set of options and their associated values and costs are sent back to the TCTA level, where the best collective set of options is chosen subject to the constraints that no aircraft can simultaneously be a member of two teams and that no target be addressed by more than one team. More detail of each of the three levels will be given in subsections 3.2.2, 3.2.3, and 3.2.4 respectively.

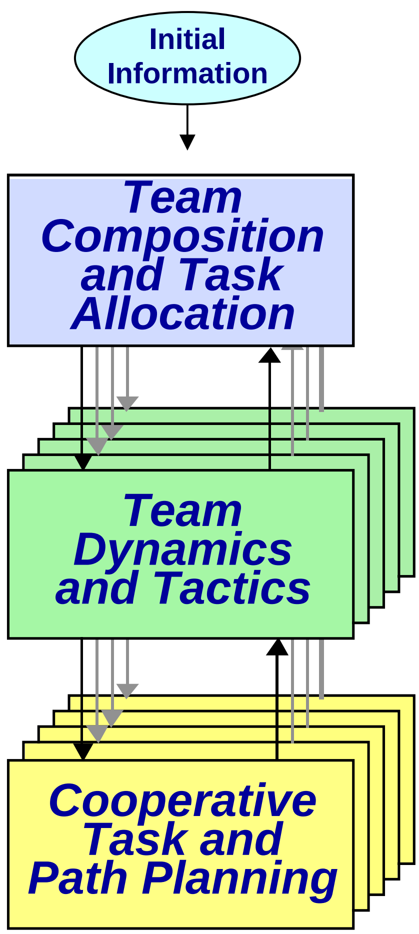


Figure 3-2: MICA Three-Tiered Planning Hierarchy

3.2.1 Initial Information

In order to start the planning process, the three-tiered hierarchy requires initial information defining the problem statement and the state of the battlespace. This initial information includes commander's intent, the current status of friendly resources, and the current status and future estimation of enemy targets.

3.2.1.1 Commander's Intent

Commander's intent is the term associated with what the commander wants to accomplish or prevent from happening in the given scenario. The commander's intent is characterized through a specification of importance of targets by time, by region, by type and with a specified level of allowable risk (cost).

- **Time** - Time represents both time phase importance as well as target time criticality (i.e. it is imperative to hit a certain target before carrying out other particular missions).
- **Region –** The commander can distinguish the importance of targets by where in the battlespace the targets reside. For example, targets in the northwest region might be more valuable than targets in the rest of the battlespace.
- **Type –** The commander can specify that certain target types are more valuable than other target types. For example, they might wish to assign more value to a surface-to-air missile site (SAM) than to a truck.
- **Allowable Risk –** The commander's intent is used to relay information regarding the importance of achieving objectives versus the loss of resources, including human.

In order for commander's intent to be employed in the MICA system, it must be mapped into quantitative values. The approach taken in MICA consists of representing the information in the following three matrices:

- ***Awareness Intent Matrix (AIM):*** This matrix includes the value of gaining awareness of enemy POIs in order to support activities other than strike (e.g., routing of aircraft around threats, tracking ground force movement) [2].

- ***Damage Intent Matrix (DIM):*** This matrix captures the specific value of damage assessment on a target beyond just determining whether additional value can be obtained by striking it again. For example, doctrine may impose specific Battle Damage Assessment (BDA) requirements on certain targets [2].

- ***Kill Intent Matrix (KIM):*** This matrix contains the value achieved for destroying an enemy target. Sensor looks that improve the effectiveness of strike activities accrue marginal KIM value [2].

In addition to valuing situation awareness and target destruction, the system also considers costs and constraints when creating plans. Some of the major components of costs and constraints are.

- ***Aircraft Value Matrix (AVM):*** Cost of losing an aircraft by type [2].
- ***Time-Sensitive Targets:*** To emphasize both the increased value and time criticality of time-sensitive targets, the commander can define an additional time-varying multiplicative factor for the value of specific Red entities.

3.2.1.2 Initial Resource Information

The resources available to the MICA system are UAVs. There are five primary types of UAVs: *large sensor*, *small sensor*, *large weapon*, *small weapon*, and *small combo*. Each aircraft platform type has different possible configurations of sensors and weapons. For example, "weapon" aircraft can only carry weapons, "sensor" aircraft can only carry sensors, and "combo" aircraft can carry both weapons and sensors. The adjectives small and large refer to how many weapons or sensors the aircraft can carry. For example, a *large weapon* aircraft can carry twenty weapons while a *small weapon* aircraft can only carry eight.

Information about these aircraft include their current locations, the amount of fuel the plane is carrying, the types of countermeasures that are affixed, the potential configuration (i.e., which weapons, sensors, etc could possibly be installed), the number of each aircraft type available, and an importance level reflected through the cost associated with each (how much is the plane worth to the user).

3.2.1.3 Intelligence Preparation of the Battlefield

Intelligence preparation of the battlefield (IPB) includes estimates about the enemy's location and identification (ID). These are estimates because the locations and ID's are not known with certainty. Each enemy resource is given a probability of being a certain type, and locations are characterized using a normal distribution. The damage state of each enemy resource is also uncertain with each categorized into one of three discrete states of damage: destroyed, damaged, undamaged.

3.2.2 TCTA – Team Composition and Task Allocation

The problem definition contained in the initial information described above is sent to the highest level of the planning hierarchy: *Team Composition and Task Allocation* (TCTA). The TCTA level has three main goals/objectives. The first is to partition the enemy targets and aircraft resources into team-sized sets. An algorithm first creates clusters of enemy targets or expected target locations. Once the enemy targets are clustered, appropriate teams of aircraft resources are assigned to address each of the clusters. This assignment process is based on how well the aircraft capabilities (i.e., sensors and weapons) match the needs to prosecute the targets in the cluster. More is discussed on the aircraft assignment process in Section 3.3.2. The goal is to maximize the value minus cost. Value is generated by destroying targets, eliminating threats, or investigating areas of interest as specified by the three intent matrices discussed in Section 3.2.1.1. Cost is computed as a function of Blue loss, Blue detection, and cost of resources. Both the values and the costs are initially entered into the system through the *Commander's Intent*.

The activities of the teams of aircraft in each resource-cluster option represent a *composite variable*. Composite variable modeling [5] is an approach to addressing the complexity of solving large-scale optimization problems. In composite variable modeling, a single decision variable is composed of a set of "traditional" or atomic decision variables, aggregated in a manner that improves the tractability of problems. For example, rather than have a variable for every possible combination of aircraft assignments to targets, variables at the TCTA level represent assignments of *teams of aircraft* to clusters of targets. However, there are a combinatorially large number of possible composites – that is, if there are T targets and A aircraft, then the total possible number of options is the cross product of all combinations of T targets and all combinations of A aircraft. Let ***nCk*** equal the number of combinations of ***n*** things taken ***k*** at a time. Thus, the total number of possible groupings of ***a*** aircraft is $\sum_{k=1}^{a} aCk$. Similarly, the total number of possible groupings of ***t*** targets is $\sum_{k=1}^{t} tCk$. The total number of possible resource-cluster pairs is the product of these two ($\sum_{k=1}^{a} aCk \;\; * \;\; \sum_{k=1}^{t} tCk$). Therefore, for a scenario including 15 aircraft and 100 targets, there are 1.27E^{30} combinations of targets and 32,767 combinations of aircraft. This results in a total number of 4.15E^{34} possible resource-cluster pairs! Our objective is to choose a small number of options and hope that among the small number are ones that will combine to form a near-optimal plan. However, if these composites are chosen poorly, the resulting detailed solution created by the lower levels will be suboptimal. Composite variables are discussed in Chapter 4.

In addition to assigning aircraft teams to particular clusters, each team is also assigned regions of interest which, based on IPB, are expected to contain additional targets. These regions of interest and the enemy targets are collectively known as Points of Interest (POI). The resource-cluster options are passed to the lower levels of the hierarchy where a detailed plan is created for each option. At the conclusion of the algorithm planning process, these detailed plans are sent back up to the TCTA level for option selection. More detail on how option selection is accomplished in given in Section 3.3.5. TCTA selects and schedules among the various team options in order to maximize *Value – Cost*.

3.2.3 TDT – Team Dynamics and Tactics

The next level in the MICA planning hierarchy is *Team Dynamics and Tactics* (TDT). As described in Section 3.2.2, the output from the TCTA level is a collection of options which consist of aircraft resources and enemy target clusters. These options are the initial input for the TDT level. The *Team Dynamics and Tactics* level adds additional plan detail to these options in three ways. First, TDT determines the optimal action to be taken for each POI within its assigned cluster. The choice of actions is constrained by the capabilities (sensing, weapons, jamming) of the aircraft paired with that cluster. The possible courses of action for each POI include any combination of looking (sensing) and striking (attacking). After the course of action is decided for each POI, the next step is to select the exact equipment (sensor, weapon, etc) to use to complete the desired action. Finally, the TDT level calculates the optimal sequencing of the POIs contained within each cluster. The choice of POI sequencing can make the difference between successfully destroying all targets in a cluster and having Blue resources destroyed because they were subjected to a large amount of risk. This sequencing is constrained by an analysis of coverage and precedence (CaP) constraints. CaP constraints are necessary for problems with targets that have overlapping threat Engagement Zones (EZ). This overlap can provide protection ("cover") by Red air defense (SAMs) for nearby Red entities in complex ways. In order to be able to prosecute a protected target safely, the threats that protect or cover that target must first be engaged and suppressed. The CaP constraints ensure that each sequence decision meets the criteria of not requiring entry into dangerous airspace (without the ability to either destroy or jam the associated target's protecting radar). For example, if the most important (i.e., highest valued) target is surrounded by a ring of lower valued enemy SAMs, precedence constraints might require destroying the surrounding lower valued SAMs before attacking the interior high valued target. Thus, the POI sequencing performed by the TDT planning level can be viewed as a traveling salesman problem constrained by the CaP constraints. Further information on coverage is given in Figure 3-4. TDT passes an ordered set of enemy targets, actions, and aircraft resources to the CTPP level for further planning refinement.

3.2.4 CTPP – Cooperative Task/Path Planning

The final level in the three-tiered hierarchy is the *Cooperative Task/Path Planning* (CTPP) level. This level adds additional detail to the plans generated by TDT for each option. CTPP determines the exact route for each aircraft in the team by minimizing route costs. The route costs are calculated as a combination of time, fuel, and the probability of attrition. In addition, this last level of the planning fills in the details required to accomplish the TDT plan. These details include sensor pointing, weapon release time and location, timing and pointing of jammers for Integrated Air Defense System (IADS) suppression, and determining decoy trajectories and time of launch. Finally, CTPP provides a detailed evaluation of the cost and value for each option that is used by the TCTA level in option selection.

Figure 3-3 gives a process flow that summarizes how the large-scale MICA optimization problem is broken into smaller more manageable subproblems as described above. The principal inputs to the process are:

1.) Commander's intent.
2.) Current best estimate of battlespace state (i.e. information on enemy targets).
3.) Available aircraft resources.

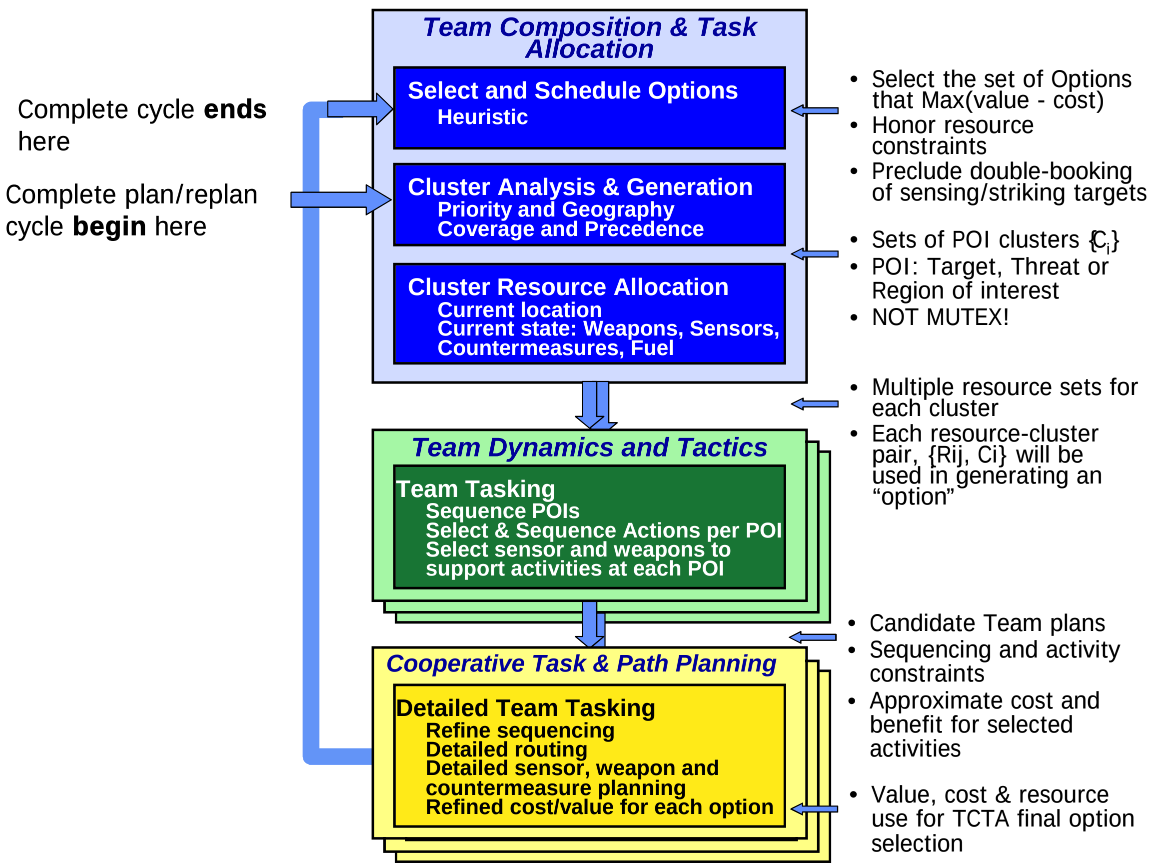


Figure 3-3: Decomposition of MICA Problem

3.3 Incorporating a Human into the System

Within the three-tiered decomposition of the planning hierarchy, there are several subproblems to be solved. Due to the complex nature of these subproblems, not all these problems can be solved efficiently by a computer. The goal of this research is to combine the inherent strengths of both a human and the computer in order to generate “better” solutions than either could produce alone. We define “better” as meaning either solutions that contain more value (closer to optimal) or solutions that are created faster but may contain the same value. With this goal in mind, we performed an analysis to determine which decisions or subproblems in MICA would benefit from Human-Machine Collaborative Decision Making (HMCDM). In addition, we attempted to establish the appropriate degree of human-machine collaboration for each of these subproblems.

We examined five of the major subproblems in MICA and identified several opportunities among them for effective HMCDM. The following subsections provide our analysis for involving a human in each of the five subproblems. However, it is important to note that only two of these subproblems were explored in further detail for this thesis: *decomposing enemy targets into clusters* and *selection of the optimal option set*. We identified these two as having the most to potentially gain by inserting a human into the problem solving process. These two subproblems are the focus of the HMCDM experiments which will be presented in Chapters 5 and 6.

As described throughout the chapter, the MICA problem is a composite variable formulation and the steps outlined above describe the process of generating and evaluating the composite variables (which have been referred to throughout this Chapter as *resource-cluster options*). It is also important to reiterate that the objective of the composite generation process is ***not*** to generate all possible options/composite variables, rather it is to generate a small set with the hope that the optimal composites (or near-optimal) will be in that set. Thus, the steps above have been designed with that in mind. Our research focuses on introducing a human into the process in an attempt to improve the quality of the options by employing the inherent human capabilities outlined in Chapter 2 in the various steps of the option generation and selection process.

3.3.1 Decomposing Enemy Targets into Clusters

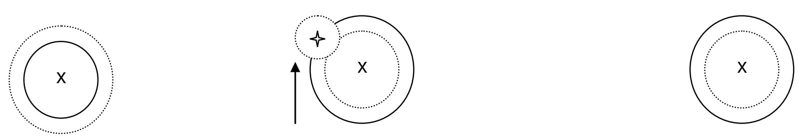
When either the initial plan or a complete replan is created, the first subproblem solved in the planning hierarchy (at the TCTA level) is the decomposition of enemy POIs into clusters. This is a critical step in that it affects each of the subsequent decisions and subproblems that follow. Because each lower level decision in the MICA planning hierarchy successively builds on the results of decisions made at higher levels, a poor choice for these initial clusters can severely degrade the quality of the final plan created. In particular, a poor choice of initial clusters could result in missions that are too risky, take too long, or achieve little value. The original design of the clustering process in MICA was a computer-only solution with no human involvement. The clustering decision was based on a set of heuristics that incorporated three main criteria: *linear distance*, *coverage and precedence (CaP)*, and *interaction*. Figure 3-4 provides more detail on the three criteria.

In addition to the three criteria, the heuristics also check to determine what class of Red entity each target in the cluster is likely to be (each target has a probability of being a certain target type due to uncertainty). Of particular interest is whether or not the POI has the potential to be a threat with the ability to shoot at the aircraft. Based on all of this information, the MICA system uses a heuristic to cluster enemy targets.

Distance – The Euclidean distance between the centers of the points of interest (POIs) or targets. This distance is calculated using the generic distance equation

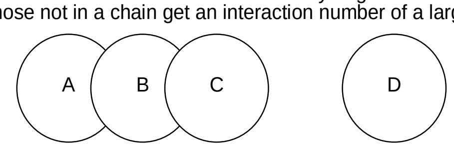
$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

Coverage – Overlapping threat Engagement Zones (EZ) can provide protection ("cover") for nearby Red entities. Therefore, each target within the problem is given a corresponding *coverage number.* Targets that are not covered at all are given a value of 0. In the first graphic below, the "dashed circle" corresponds to the target with a *coverage number* of 0. The "solid circle" target is inside, thereby not covering the "dashed circle" target. In the second graphic, the target identified by the arrow is partially covered by both the "dashed circle" target and the "solid circle" target giving it a *coverage number* of 1. In the final graphic, the "dashed circle" target is completely covered by the "solid circle" target. This represents a *coverage number* of 2.



Coverage number = 0 =1 =2

Interaction – Defines how closely targets are "related" to one another. Targets that are part of a "chain" receive an interaction number relative to how many degrees of separation exist between the two targets in question. Those not in a chain get an interaction number of a large positive constant.



A is related to A with Interaction = 0
A is related to B with Interaction = 1
A is related to C with Interaction = 2
A is related to D with Interaction = M (where M is a large positive number)

Figure 3-4: Numerical Values Used in Computer Clustering Heuristic

One shortcoming of the clustering approach is that if targets are "blocked" by other targets in less obvious ways, some targets can be left out of clusters entirely. This often leads to high-valued targets that are not included in the final plan generated by the MICA system. Figure 3-5 provides an example of this phenomenon. In the figure, although there is no explicit coverage by any of the threats on "the wall", *target 1* is still being blocked by these targets. In addition, the distance between *target 1* and any of the threats is great and the interaction numbers are all large positive numbers. All of these factors result in the heuristic not creating any clusters including *target 1*.

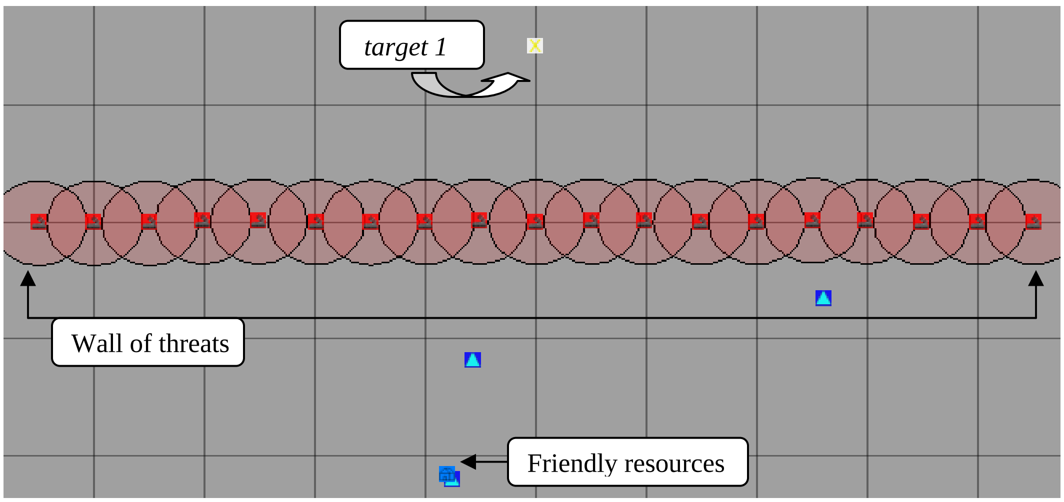


Figure 3-5: Wall of threats blocking a high value target (i.e., *target 1)*

One solution to this problem is to design a better clustering algorithm. However, involving a human in the clustering process might provide a more effective way to create better solutions. Human operators can quickly perceive the big picture and can take advantage of their ability to recognize patterns (See Figure 3-6) and can use spatial reasoning to create clusters. Spatial structure helps to reduce complexity for humans but might not for computers or algorithms. Humans are good at identifying patterns with spatial structure such as in Figure 3-7. Although there are many heuristic clustering algorithms that deal with this kind of problem, the complex nature of the interactions among threats and targets in MICA make the development of a general heuristic problematic. A human can dynamically exploit different aspects of the problem depending on the scenario while the computer is limited by its own algorithm. For example, instead of creating an exhaustive list of logic to account for every possible scenario intricacy, we can tap into the human resource to quickly identify strategies to overcome problem-specific aspects. Of course, there may be cases for which an operator would have difficulty in identifying a clustering pattern, especially for scenarios that have uniformly distributed targets such as Figure 3-8.

It is also possible that humans would have trouble clustering targets in scenarios that contain an enormous number of enemy targets. The overwhelming number of targets might be too much for a human to process especially if they cannot easily distinguish CaP constraints from the graphical display. In these situations, a computer's ability to handle larger problems and to recognize more complicated CaP constraints is useful. Therefore, it is our conclusion that target clustering should be a joint effort between a human and a computer algorithm. Even situations in which a human operator is only able to easily define a few clusters, this will narrow the large search space for the computer to cluster the remaining targets.

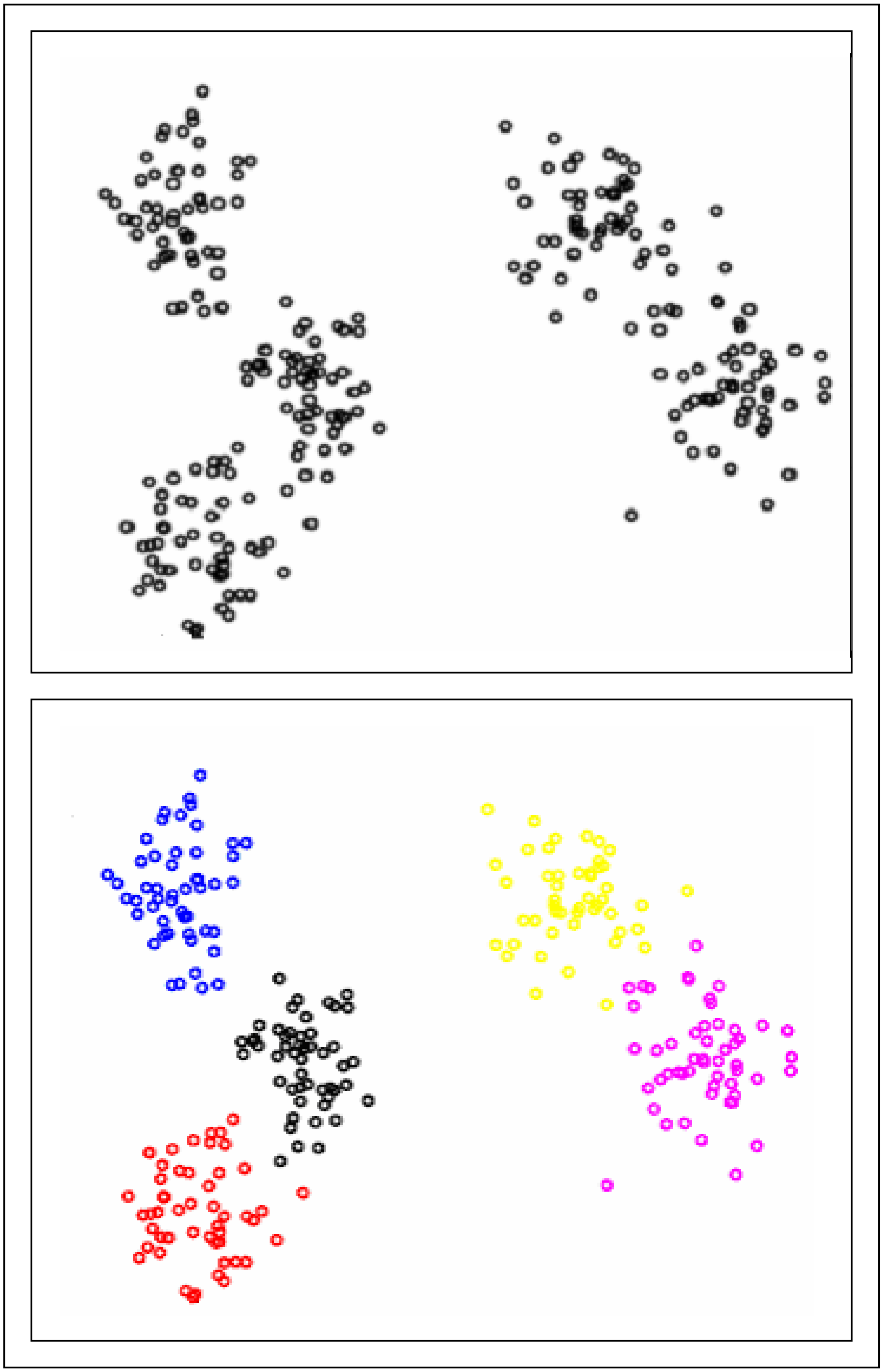


Figure 3-6: Generic Example of Exploiting Human's Ability to Visually Cluster [17]

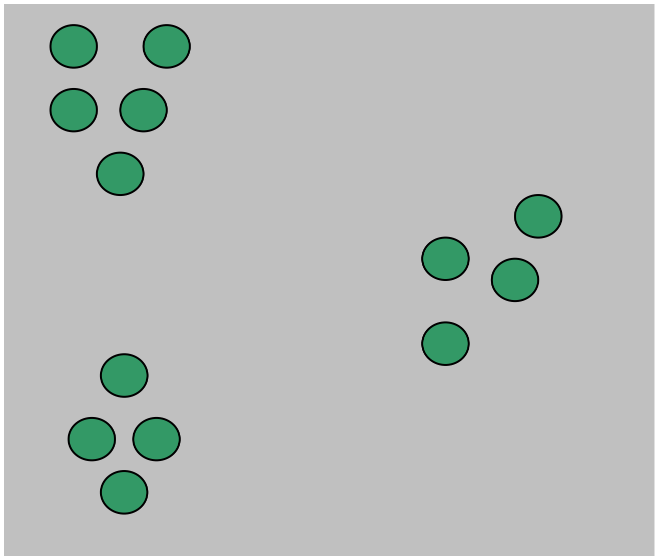


Figure 3-7: Generic Example of "Easy" Scenario for Human to Cluster

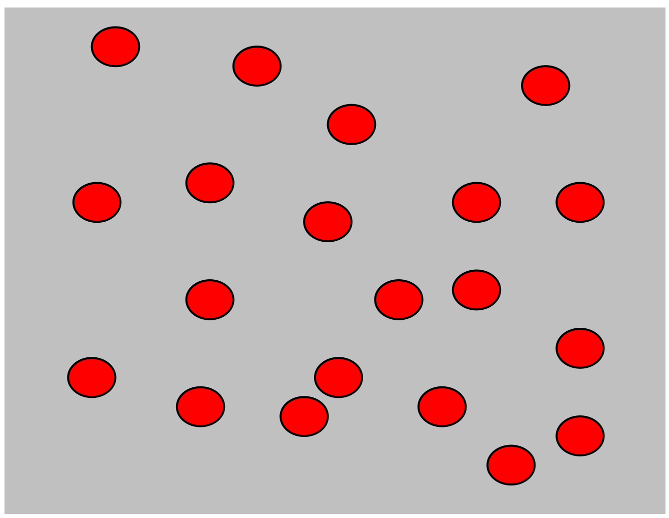


Figure 3-8: Generic Example of "Difficult" Scenario for Human to Cluster

3.3.2 Formation of Aircraft Teams

After POIs are clustered, the TCTA level assigns a team comprised of appropriate types of aircraft for each cluster. This assignment process is another MICA subproblem that may benefit from HMCDM. MICA currently uses heuristics to create three different aircraft teams for each cluster. The heuristics use scores based on the requirements for prosecuting the POIs within the cluster and the capabilities of the aircraft in forming the teams for a given cluster. The rationale behind the selection of the three team types for each cluster follows.

- *Varsity Pairing:* Assigns the best possible aircraft to be paired with each cluster. Starts by identifying a particular cluster and a general idea of what actions to take for the POIs in that cluster. For this pairing procedure, it is assumed that all aircraft are available to be chosen. The procedure starts by adding aircraft that provide the most value given the associated cluster and continues until there is a group of aircraft that can accomplish what needs to be done to the entire cluster. This process is followed for all remaining clusters until an optimal group of aircraft has been created for each cluster. Clearly this pairing procedure does not take into account the constraints on available aircraft. For example, this pairing procedure might identify eight clusters that would benefit most from having a large weapon aircraft and assign one to each cluster but there might only be three large weapon aircraft available for the scenario.

- *Junior Varsity Pairing*: This pairing procedure accounts for the number of each aircraft commodity used in the Varsity Pairing, making it less likely that an aircraft that was in high demand in the Varsity Pairing would be used in a Junior Varsity Pairing. For example, this pairing procedure might not assign a small combo aircraft to a cluster if it knows that small combo aircraft have been assigned to numerous other clusters (including those pairings created in the Varsity technique). At the end of this pairing procedure, there are options created from the Junior Varsity Pairing procedure and the Varsity Pairing procedure.

- *Mutually Exclusive Pairing*: The third and final pairing procedure is the mutually exclusive pairing procedure. This procedure rank-orders the clusters based on their potential value so that the most valuable cluster is assigned aircraft that best meet the objectives of the actions planned for the targets within that cluster. This process continues for the second highest valued cluster and so on until all clusters have been assigned aircraft. This procedure differs from the other two in that it only assigns available aircraft. For example, if there are only five small weapon aircraft available and all five have previously been assigned to clusters, the algorithm accounts for the fact that no small weapon aircraft are available and moves on to the next best aircraft type. "Best" is defined as the aircraft type whose capabilities match best to the actions to be carried out for the targets within the cluster.

Note that the number of possible combinations of aircraft assignments to clusters is extremely large and may require considerable computation in order to optimally match teams to each cluster. By involving human collaboration in this subproblem, the time to select good (or near-optimal) teams of aircraft might be significantly decreased. Based on a combination of experience, tactics and doctrine, a human might be able to immediately eliminate "bad" choices of aircraft teams. A bad choice would be a particular composition of aircraft that would not be effective or reasonable for a certain cluster of POIs. Reasons for not using particular aircraft range from certain terrain limitations that are not currently modeled in MICA to a commander's reasons for not wanting to use a certain aircraft or configuration.

In order for a MICA user to make good team composition decisions, the user might have to be trained. This task is not as simple as visually clustering items based on their spatial similarities. It requires extensive experience and knowledge of the different aircraft types and their capabilities. By combining the computer's ability to find available aircraft with the desired capabilities quickly and keep track of many different options with operator experience in pruning the range of choices, human-machine collaboration has the potential to add benefit in solving this sub-problem. It is important to note that the possible benefits of HMCDM in this subproblem extend beyond the value of the solution alone. For example, by involving a user in the assignment of aircraft, the user's confidence in the solution generated could significantly increase.

3.3.3 Sequencing of Enemy Targets within the Clusters

Another subproblem in the MICA hierarchy that could benefit from HMCDM is the sequencing of targets within clusters. The target sequencing problem in MICA is stated as a multi-vehicle traveling salesman problem that uses estimates of cost between pairs of targets in order to find the best sequencing solution. Although the number of targets in each cluster is typically small (less than 10), the number of different sequences can be very large. For example, a cluster consisting of 10 targets with no precedence constraints has 3,628,800 different ways of being sequenced. A human would certainly not excel at enumerating or evaluating all of these sequences. However, there might be a certain sequencing that is readily identifiable by a human based on their experience, intuition or current tactical doctrine. They might also be able to reduce the number of sequences for the machine to evaluate by quickly identifying undesirable full or partial sequences.

3.3.4 Individual Aircraft Routing

After the targets have been clustered, teams of aircraft assigned, actions on targets decided, and the sequencing of actions determined, the next subproblem to be solved deals with the detailed routing of the individual aircraft between POIs and the actions at a POI. The current method uses an A* search [1] over a regular grid of the battlespace to calculate the shortest (least costly in terms of an objective function that trades off time and risk) paths between POIs. The results depend upon the size of the grid used: the smaller the grid size, the longer the computation will take, but safer / shorter routes are more likely to be found.

The computer excels at this sort of well-defined search, and can keep track of the fuel usage, time usage, and accumulated probability of attrition for any path generated. Because of the large number of possible routes and the amount of information to be kept for each route, a human would have a difficult time solving this problem without some computational support from the machine. However, there are several advantages to involving a human in this task: a user could take into account features of the terrain, specify particular waypoints, specify weapon release points, or even see paths which the computer perceives as blocked, due to the aforementioned shortcoming of searching over a regular grid. By doing so, the human could quickly decide portions of routing, thereby decreasing the amount of required computer calculation. Although this subproblem showed potential for HMCDM involvement, it was not explored in detail for this thesis. The expected benefit was not anticipated to be significant relative to the effort required to enable human interaction.

3.3.5 Selection of the Optimal Option Set

After the TDT and CTPP levels add plan details to each of the resource-cluster options originally created in the TCTA level, the full options along with their cost and value are sent back to TCTA for option selection. The option selection problem is an integer programming problem [2] that is currently solved by a heuristic. The original MICA design envisioned a “pool of tens of thousands of options" [1]. However, due to the computational complexity of the option generation process as described above, the number of options created is typically in the tens. The fact that there are a small number of options is key in assuming that human involvement might add significant benefit in option selection. The process might entail the computer calculating and displaying metrics for each of the options and the human evaluating these metrics in selecting which options should be contained in the final solution.

The subjective nature of the solution quality makes it difficult for a computer to quantify if one option is “better” than another and if it is, to what degree is it better. By involving a human in the process, they are able to make the risk-reward trade-off and dynamically determine which metrics are most important. For example, in some situations, the time to conduct the mission might be the most important factor whereas in other situations the number of resources used might be the most important. The human could tap into their experience, intuition, and understanding of the big picture to conduct the trade-offs instead of being forced to quantify time or relate numerically how important *resource usage* is compared to the *value* of the solution.

Even if the MICA system were modified to actually produce pools of thousands of options, humans could still add benefit to the option selection process. However, the way in which humans were involved would have to change in order to stay effective. It would no longer be a possibility to present a human with information about thousands of different options at once. This amount of information would overload the human and most likely not result in optimal solutions. However, instead of attempting to evaluate all the options at once, the humans would be given smaller sub sections of options at a time. Of these smaller sub sections, humans could identify options they definitely wanted placed in the final solution, or options to be "held onto" and carried over into the next iteration for further evaluation.

Human involvement could also add benefit even if an Integer Program solver were employed in the MICA problem. An IP solver would pick the best set of options subject to the constraints that no aircraft is used in two different options and no POI is addressed in two different options (i.e., no aircraft and no POIs are "double-booked"). In this situation, we expect that a human operator would be well used in an iterative process. After the IP solver found the best solution among the options generated, the operator could inspect that solution and identify how unused aircraft and unused targets might be used in creating new options or in modifying existing options. This would ensure that as many aircraft as possible are used and as many POIs as possible are addressed. This idea of an *Iterative Composite Variable Approach* is discussed further in Chapter 4.

In addition, there is a great amount of benefit in having the human feel involved and understand the solutions that are being created. Because this is the last step in the plan creation process, human involvement in this step is crucial for establishing trust in the final solution.

Chapter 4

Large Scale Optimization -- Goal Decomposition & Composite Variable Formulation

The development of algorithms to solve complex, large-scale, optimization problems, poses a number of significant challenges. Here we consider the class of problems whose decision variables represent activities over time that are traditionally formulated as Integer Programming (IP) problems. Because of the large scale of these optimization problems, the IP formulation requires an enormous number of variables in order to represent all possible decision alternatives. For instance, the problem of optimizing the scheduling and routing of 20 aircraft across a network of 100 locations over a time interval of one day discretized into one-minute periods requires 2.88 million integer variables [1]. In addition to the large number of decision variables, there are also a combinatorial number of constraints required for this problem. These constraints ensure that aircraft are not scheduled to fly faster than they are physically capable and that they are not in two places at the same time. Adding to the complexity of the solution to these problems is the fact that these constraints induce considerable fractionality in the linear programming relaxation solutions. This fractionality results in solution times that are typically exponential in the number of integer variables. In summary, the massive number of variables and constraints coupled with the fractionality makes it nearly impossible to solve such problems using traditional IP approaches.

Two techniques which have been found to be useful in overcoming the difficulties associated with such complex, large-scale, optimization problems (such as the MICA C^2 problem presented in Chapter 3) are *composite variable formulations* and *goal decomposition.* This chapter describes, compares and contrasts these two formulation and solution methododologies. We discuss their respective strengths and weaknesses in the context of addressing complex large-scale optimization problems. Finally, we outline a proposed strategy for incorporating both methods in an HMCDM context. We call this strategy the *Iterative Composite Variable Approach*. We end the chapter by describing how this approach can be used in the MICA application to generate "better" solutions. A "primitive" version of the Composite Variable-HMCDM approach, in which only one iteration was employed, was used to obtain the results presented later in this thesis.

4.1 Linear and Integer Programming

We begin our discussion of *composite variable formulations* with a brief overview of the application of traditional Linear Programming (LP) and Integer Programming (IP) approaches to scheduling and resource allocation problems.

Mathematical models are used to describe linear programming problems. All mathematical functions in an LP are required to be linear functions of the decision variables. The word programming is essentially a synonym for planning. Thus, linear programming can be viewed in the context of the problems of interest to us as the planning of activities to obtain an optimal result among all feasible alternatives [16]. Linear programming can be used to allocate limited resources optimally among competing activities. The problem involves selecting the levels of activities that compete for scarce resources. The choice of activity levels dictates how much of each resource will be consumed by each activity. Linear programming can be applied to a variety of different situations. However, in each of these situations, the common ingredient is the necessity for allocating resources to activities by choosing the levels of those activities.

Although allocating resources to activities is the most common application of linear programming, it has numerous other important applications as well. In fact, *any* problem whose mathematical model fits the very general format for the linear programming model is a linear programming problem [16].

In many practical problems, the decision variables of the problem are limited to taking on integer values. For example, if the problem is to decide how many stores to build in locations around the U.S., a solution that calls for 0.6 stores in one location and 0.7 in another location would not make sense because it is not possible to build fractions of a store. In such cases, an integer program is used instead of a linear program. The IP mathematical model is the LP model with the additional constraint that all decision variables take on integer values.

The computational effort required to solve integer programs depends a great deal on the problem structure. IP solvers generally solve a sequence of linear programs in a branch-and-bound tree search that terminates upon finding an integral solution [24]. The more naturally integral the underlying linear program is, the less amount of work required of the generally exponential-performance branch and bound algorithm. Therefore, problems whose linear relaxations naturally produce integral or nearly integral solutions solve rapidly in IP solvers. However, in most real life applications, solutions are not naturally integral resulting in a significant amount of computation time and effort needed to solve the problem.

4.2 Composite Variable Approach to Solving Large-Scale Optimization Problems

Composite variable modeling is a recently developed approach to solving large-scale optimization problems that attempts to reduce the fractionality which is often present in real life problems. The concept was developed in [5] and has been further applied in several applications, including [8], [13] and [23]. In composite variable modeling, a single decision variable is composed of several "traditional" decision variables. A typical example of a composite variable is a binary decision variable that represents whether or not to implement an entire set of decisions. The advantage to using this approach for large-scale problems is that it can lead to less fractionalization in the solution which leads to a strong linear programming relaxation. In other words, the solutions to the linear programming relaxation of the composite variable problem tend to be more nearly integral, resulting in solutions of the integer program with fewer iterations of the branch-and-bound technique. Problems formulated using composites also have significantly fewer constraints because many of the constraints of the original problem become captured in the composite variable formulation. However, due to the numerous ways of combining traditional variables to form composites, composite formulations generally lead to an increased number of integer variables. An example showing this effect in the MICA problem is given in Chapter 3 (Section 3.2.2) and is summarized here. If there are

T targets and A aircraft, then the total possible number of options is the cross product of all combinations of T targets and all combinations of A aircraft. For a problem with 15 aircraft and 100 targets, there are 4.15E^{34} possible composite variable combinations. The original problem would also have all of those combinations to deal with as well as the details of the missions (e.g., the trajectories of the aircraft, etc). Thus, although the composite formulation is complex – the original problem is even more so.

The large number of variables is typically overcome by applying application-specific rules/heuristics to reduce the number of composites that are considered. For instance, in the 15 aircraft, 100 target example given above, it is likely that not all of the aircraft would be able to effectively prosecute all of the targets. This may be because some of the aircraft are not equipped with the appropriate weapons or sensors. In general, composites that "do not make sense" or are *dominated* (see Section 4.2.1.2 for further discussion on dominated composites) by other composites are discarded from consideration. Thus, the challenge in composite variable formulation is the composite generation. The objective of the composite generation process is ***not*** to generate all possible composite variables, rather it is to generate a small set with the hope that the optimal composites (or near-optimal) will be in that set.

4.2.1 Example of the Benefit of Employing Composite Variables

Unfortunately, most natural IP formulations of the type of battle management problems encountered in MICA are not readily solved, even with the latest IP solvers [2]. For this reason, a composite variable formulation proves extremely beneficial in solving these types of problems. To illustrate the effectiveness of such an approach, we examine a simple delivery problem [8] and compare the formulation of the problem using traditional integer programming with the equivalent composite variable formulation.

4.2.1.1 Integer Programming Formulation of Problem

Consider the delivery optimization problem defined on the network shown in Figure 4-1 [1] [8]. The network has seven time periods and five physical locations. The time periods are displayed in the columns in the form of numbers from one through seven. The five locations are listed in the rows: base1, base2, tank1, tank2, and targ1. Tank1 and tank2 represent tankers which can refuel our aircraft in flight. Arrows indicate the feasible movements of aircraft in this time-space network. For instance, the arrow from base1 at time period one to base1 at time period two indicates that an aircraft may stay at base1 from time period one through time period two. The other option for an aircraft at base1 in time period one is to travel to base2, which is notated by the remaining arrow pointing out of base1 at time period one.

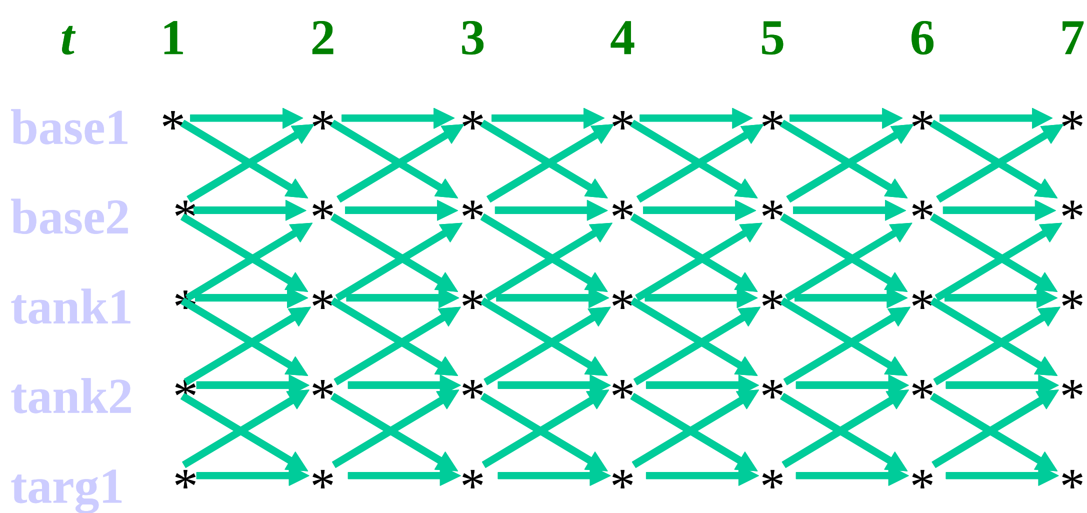


Figure 4-1: Network Representing a Delivery Problem with 5 Locations and 7 Time Periods

The decisions in this delivery problem are the movements of aircraft in each time period to one of five physical locations. The objective is for the aircraft to "deliver weapons" or strike the enemy target (targ1) in the most efficient manner. These decisions are represented by binary variables which are a special case of integer variables in that they are restricted to the integer values 0 and 1. Binary variable formulation is used in problems with "yes" or "no" decisions. Thus, state $\boldsymbol{i}$, state $\boldsymbol{j}$ and time $\boldsymbol{t}$ can be represented by the decision variable $\boldsymbol{x_{ijt}}$ such that

$$x_{ijt} = \begin{cases} 1 \ \textit{if aircraft moves from state } \mathbf{i} \textit{ to state } \mathbf{j} \textit{ in time period } \mathbf{t} \\ 0 \ \textit{otherwise} \end{cases}$$

This move is made with the associated cost $\boldsymbol{c_{ijt}}$. The cost might represent any number of different factors to include the cost of aircraft attrition, fuel, etc. Otherwise, $\boldsymbol{x_{ijt}}$ is set to 0 ("no"), representing that the aircraft *will not* move from state ***i*** to state ***j*** at time ***t***. Therefore, the objective function of a relevant IP might be:

$$\min \sum_{ijt} c_{ijt} x_{ijt} \tag{4.1}$$

A large number of constraints is required to ensure that the same number of aircraft entering each location also leaves it (conservation of flow constraints):

$$\sum_j x_{jit} - \sum_j x_{ijt+1} = 0, \forall i, t \tag{4.2}$$

In addition, there may be additional constraints to ensure that there are enough aircraft to carry the required payload (enough supply to meet demand constraints):

$$\begin{aligned} &\sum_j w_{ijt} - \sum_j w_{jit} \le sply_{i,t} - demand_{i,t}, \forall i, t \\ &w_{ijt} \le (cap) x_{ijt}, \forall i, j, t \end{aligned} \tag{4.3}$$

where $\boldsymbol{w_{ijt}}$ represents the weapons flown from ***i*** to ***j*** at time ***t***, and ***cap*** represents the weapons capacity of an aircraft. The second constraint in Equation 4.3 is a capacity constraint that ensures that the amount delivered does not exceed the carrying capacity. These constraints can cause fractional solutions if the capacity of an aircraft is not exactly equal to the demand of the delivery points. According to the objective function, it is cheaper to fly a fraction of an aircraft than to fly a whole one. This fractionality in the linear relaxation requires large numbers of branch-and-bound iterations in the corresponding IP.

4.2.1.2 Composite Variable Formulation of Problem

The equivalent composite variable formulation is depicted in Figure 4-2 [8]. A1, A2 and A3 are three unique aircraft types and the corresponding trajectories from the aircraft symbols represent the path the aircraft flies for the duration of its mission. In the example, an A3 type aircraft begins at base1 in time period 3, flies to and attacks targ1 in time period 4 and then returns to base1 in time period 5. This collection of decisions is a composite variable or option. There are five composite variables for this example, one for each of the aircraft shown in Figure 4-2.

Rather than require the IP to solve the details of aircraft routes and loads, the composite variable formulation delegates that task to a large number of subproblems, in a form of hierarchical decomposition which was discussed in Chapter 3.

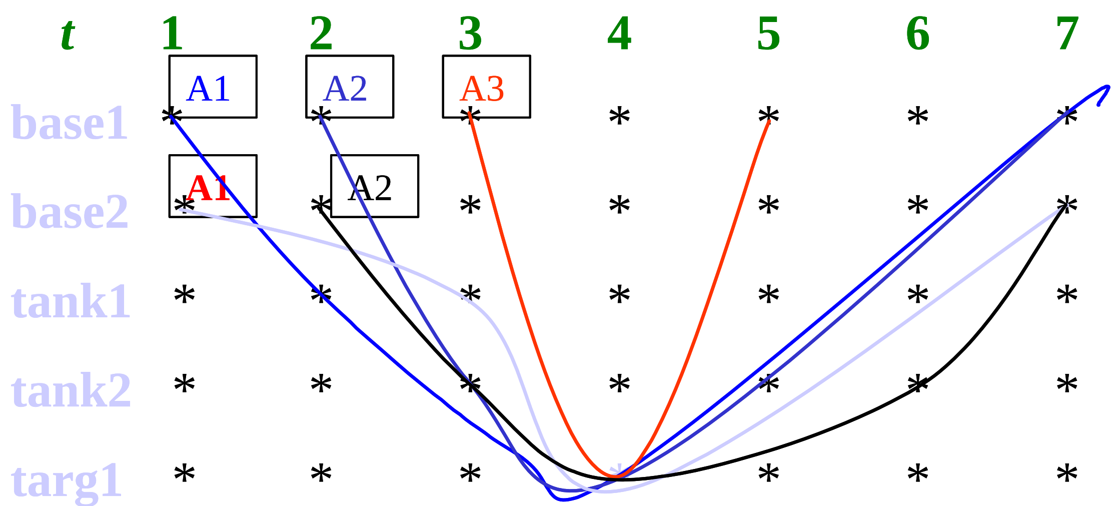


Figure 4-2: Composite Variables Defined for the Same Delivery Problem

The sub-problems are responsible for producing routes and aircraft loads that are *potentially* part of the globally optimal solution. In other words, the subproblems generate the details of the plans for the pool of options or composite variables that will be used in selecting the final solution. The characteristics of each route required by the IP are captured in coefficients produced by the subproblem [1]. In this example, the only characteristic from the subproblem needed by the composite variable master problem is the cost c_i associated with route i. The decision variables in the composite variable master problem are whether or not to choose the entire collection of decisions represented by each composite. The decision variables z_i are set to 1 when the corresponding composite **i** is selected, and set to 0 otherwise. The resulting integer program is the following [1]:

$$\text{Minimize } \sum_i c_i z_i$$

$$\begin{aligned} \text{Subject to: } & \sum_{i \in Is} z_i \geq 1, \forall\, Is \\ & \sum_{j \in Js} z_j \geq 1, \forall\, Js \\ & z_i \in \{0,1\}, \forall i \end{aligned} \tag{4.4}$$

where $\boldsymbol{I_s}$ is associated with covering constraints requiring that at least one of the options in each set is chosen, and $\boldsymbol{J_s}$ is associated with requiring that at most one way to accomplish each objective is chosen [1]. The structure of this binary program, with all coefficients 0 or 1, is far simpler for the IP solver to handle. However, the drawback as mentioned in Section 4.2 is that depending on the specifics of the problem, the composite variable formulation might involve a very large number of variables. As seen here, the composite variable approach can combine many inter-related decision variables (in this example, aircraft loads and aircraft routing decisions) into sets of variables. This avoids the fractionality aspects which are difficult for an integer program to solve.

The composite variable formulation approach has proven successful in solving a variety of complex real-world problems. UPS upper management conservatively credits $87 million savings over the last three years to the use of a composite variable formulation to solve their overnight delivery route design problem [6]. Composite variable approaches have also significantly improved solutions in military and delivery problems [7], [8], [13] and [23].

However, the aforementioned successes were for problems with specific structure that allowed the set of composite variables $\boldsymbol{z_i}$ to be pre-calculated and stored within conventional memory limits [1]. Their structure allowed the identification and calculation of a relatively small set of *dominant* composite variables a priori. We define a dominant composite variable to be one that represents a collection of decisions that is known to be superior to other sets of decisions. These dominant composites eliminate the need to further create or consider any of the *dominated* sets of decisions.

For example, in the composite variable formulation shown in Figure 4-2, suppose there was a composite variable representing a strike of targ1 using two small aircraft subjected to a low amount of risk. For this to be a feasible composite, the two small aircraft would need to be able to carry enough weapons to carry out the strike. This composite would dominate any other combination of aircraft chosen to strike the same target if their mission included flight routes exposed to higher risk or requiring more time to carry out the strike or more than two aircraft to supply the same amount of weapons to conduct the strike.

Within the more complex battle management problems that the MICA hierarchy must solve, the set of dominant composite variables, $\boldsymbol{z_i}$, is too large to exhaustively generate. Because an exhaustive list of composite variables is not produced, the key to solving the problem then becomes how to generate the best possible pool of composite options. This is not as trivial as trying to generate individually good options. We want the *collection* of options that as a *whole* provide the *overall* optimal solution. For example, even if one option within the pool is extremely good, it might have numerous overlapping constraints with other options whereby these other options within the pool can not be included in the solution. This suggests that the global solution could be better with a collection of two not quite as good composites that don't use as many resources or have as many overlapping constraints with other options.

4.3 Hierarchical Decomposition (Multi-Level Optimization)

The objective of hierarchical decomposition (multi-level optimization) is to decompose a complex optimization problem into a hierarchy of smaller, more easily solvable subproblems whose solutions combine in a way that retain the original objective and constraints of the complex problem [10] [3]. The simpler optimization problems are solved separately at each level of the hierarchy. Throughout this process, solutions at the higher levels produce objectives and constraints that are used by lower levels in a way that ensures optimality and, depending on the decomposition method, feasibility. More specifically, the higher levels of the complex problem coordinate the solutions of the decoupled lower level problems through the use of coordinating variables. The original problem is solved by a *master problem* that sets the coordinating variables used within the subproblems. The solutions to the subproblems at the lowest levels of the hierarchy represent a plan of activities that is pursued by physical entities to prosecute the goals of the larger original problem [1]. In the context of the MICA system used in this thesis, the physical entities are UAVs and the plans of activities are individual strike and sensing actions on enemy targets. Hierarchical decomposition, when used appropriately, can reduce solution times dramatically with little or no loss of plan effectiveness.

4.3.1 General Approach

Consider the following problem statement [3]:

$$\min_{x,y} f(x,y) \quad \textbf{subject to} \;\; g(x,y) \le 0 \;\; ; \quad \textbf{where} \quad x = [x_1 \; x_2 \ldots x_N]^T \tag{4.5}$$

The vector $\boldsymbol{x}$ is composed of N subvectors $\boldsymbol{x_i}$ which are the decision variables associated with each of the subproblems at the lower levels. The vector $\boldsymbol{y}$ corresponds to the decision variables that couple the subproblems through the objective function $\boldsymbol{f}$, the constraint vector $\boldsymbol{g}$ or both. These variables are referred to as the *coordinating variables*. The original problem can be rewritten in terms of a Lagrangian $\boldsymbol{L}$ with Kuhn-Tucker multiplier vector $\boldsymbol{\gamma}$ [3]:

$$L(x,y) = f(x,y) + \gamma^T g(x,y) \tag{4.6}$$

The decomposition of the original optimization problem is achieved by creating an upper level problem referred to as the master problem. When the master problem sets the value of the coupling vector $\boldsymbol{y}$**,** the decomposition approach is referred to as *interaction prediction* or *goal (feasible) decomposition*. When the master problem sets the value of the multiplier $\boldsymbol{\gamma}$, the decomposition approach is referred to as *price decomposition*. Given the values for the coupling variables, the Lagrangian can be rewritten as a sum of decoupled Lagrangians $\boldsymbol{L_i}$. More specifically, setting a value for $\boldsymbol{y}$ leads to separability of both the objective function $\boldsymbol{f}$ and the constraint vector $\boldsymbol{g}$. This separability leads to the following formulation [3]:

$$L(x,y) = \sum_{i=1}^{N} \left[f_i(x_i, y) + \gamma_i^T g_i(x_i, y) \right] = \sum_{i=1}^{N} L_i(x_i, y) \tag{4.7}$$

Once the master problem sets the values for $\boldsymbol{y}$, each of the N lower levels is responsible for solving a decoupled optimization subproblem associated with a sub-Lagrangian $\boldsymbol{L_i}$. Iterations between the upper and lower levels are required to achieve an optimal solution.

$$\min_{x_i} f_i(x_i, y) \qquad \textbf{subject to} \;\; g_i(x_i, y) \le 0 \tag{4.8}$$

The most important characteristic of the hierarchical decomposition technique is that the system-wide objective function and constraints are reflected in the solutions to the subproblems.

4.3.2 Goal Decomposition

Goal decomposition ensures that the solution arrived at each iteration is *feasible* (satisfies all constraints). In goal decomposition, the coordinating variables are used to break the problem down into independent smaller subproblems with sets of resources and sets of objectives. In an iterative process, the lower levels create solutions to their individual problems as well as sensitivity of those solutions to the values of the coordinating variables set by the master level. Given the "answers" to the subproblems and the sensitivity information, the master level resolves for the coordinating variables. Iterations continue until diminishing returns is achieved. The sensitivity information can be viewed as providing insight to the master level as to how to redistribute resources or objectives to the lower levels. For example, this might involve taking away resources from one and giving them to another, or redistributing objectives. One of the keys of goal decomposition is that it ensures a feasible solution at each iteration. Goal decomposition also ensures that an optimal solution will be found as long as there is no restriction on the number of iterations. However, depending on the nature of the problem it may be difficult for the subproblems to develop the values of the sensitivities that are used by the master level in guiding its iterations.

4.4 Comparison of Composite Variable Formulation and Goal Decomposition

Goal decomposition and composite variable formulations have many similarities in the way that they are used to simplify large-scale complex problems. Goal decomposition breaks down the larger problem into smaller, more readily solvable subproblems. Similarly for the composite variable formulation each composite can be viewed as a subproblem. An "option" or composite is much the same as one of the lower level problems defined by the master level in the goal decomposition context. That is, a composite is basically a subproblem that is defined (by some possibly heuristic mechanism) and is solved (i.e., aircraft missions to attack targ1 in Section 4.2.1.2) by some lower level optimization routine. In the goal decomposition approach, the solutions to the lower level problems are passed back "up" to the master problem. In the composite variable approach the value and cost (basically the solution without details) of the defined options/composites are passed back to another (higher) level that uses an IP to select among the composites in order to achieve the best overall solution that satisfies the constraints. A difference between the two is that the goal decomposition lower level solutions provide, in addition to the subproblem solutions, resource sensitivities back to the master problem. These sensitivities are used to reallocate resources in a manner that produces a higher valued solution. Goal decomposition is an iterative process whereby a feasible solution is obtained at each iteration and the optimal solution is eventually reached. Composite variable formulations as defined in [5] do not include resource sensitivities and also do not necessarily involve an iterative process. Typically, a composite variable formulation includes an a priori pool of all generated composites or options. Based on this pool of composites, an IP solver selects the best composites subject to the constraints of the problem. There is no guided iteration to improve the composite pool in an effort to reach an optimal solution in composite variable formulation, and therefore the quality of the solutions relies solely on the quality of the initial pool of composites generated. Another contrast of the two is that in goal decomposition the collective lower level solutions always combine to yield a feasible solution to the overall problem. In the composite variable approach, the lower level problems "overlap" in the sense that some of them may be addressing the same objectives or in that some of them may be using the same resources – the higher level IP sorts through the composites to choose the set that yields the most value while satisfying objective and resource constraints. Figure 4-3 and Figure 4-4 provide a graphical depiction of composite variable formulation and goal decomposition respectively.

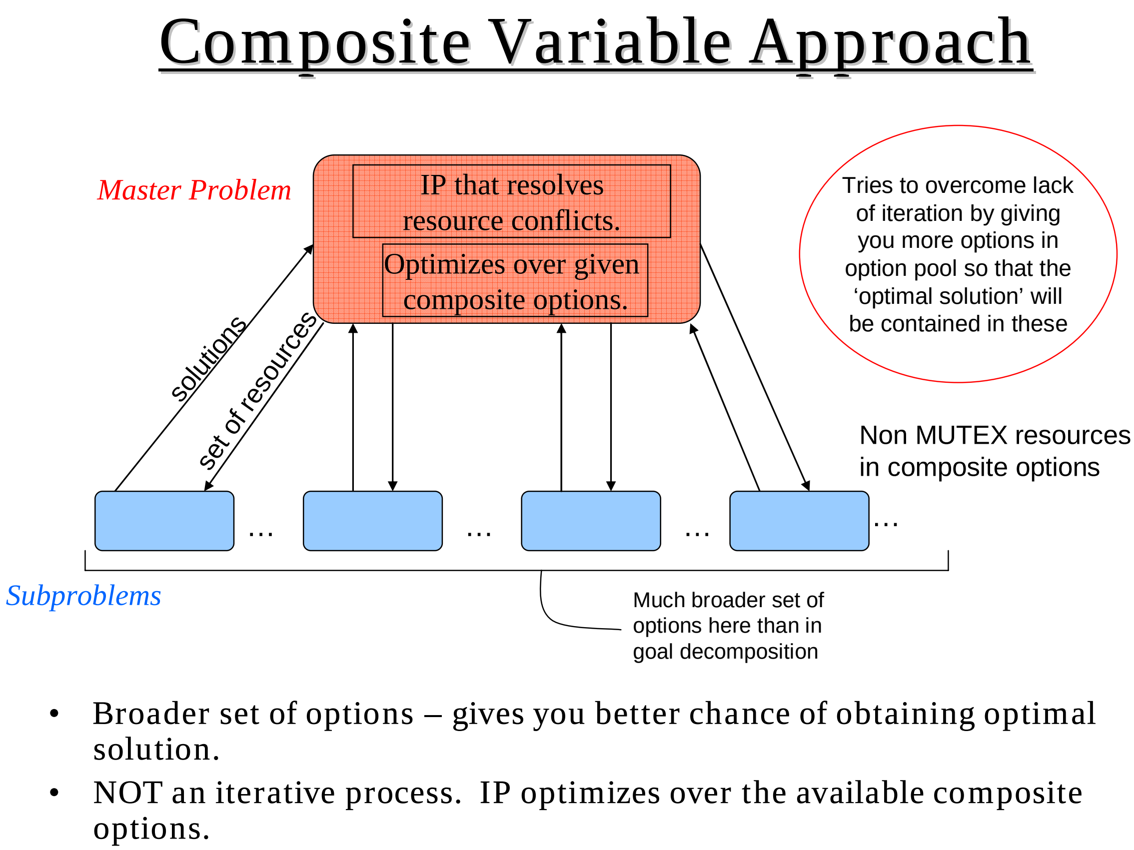
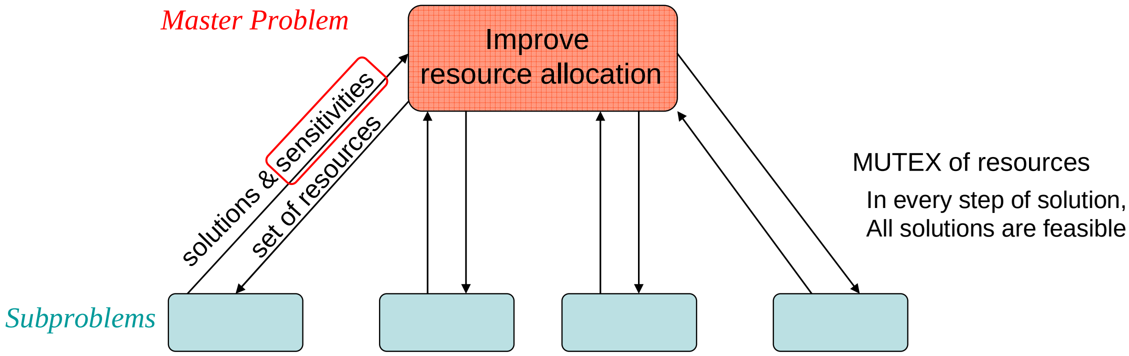


Figure 4-3: Composite Variable Approach

Goal Decomposition



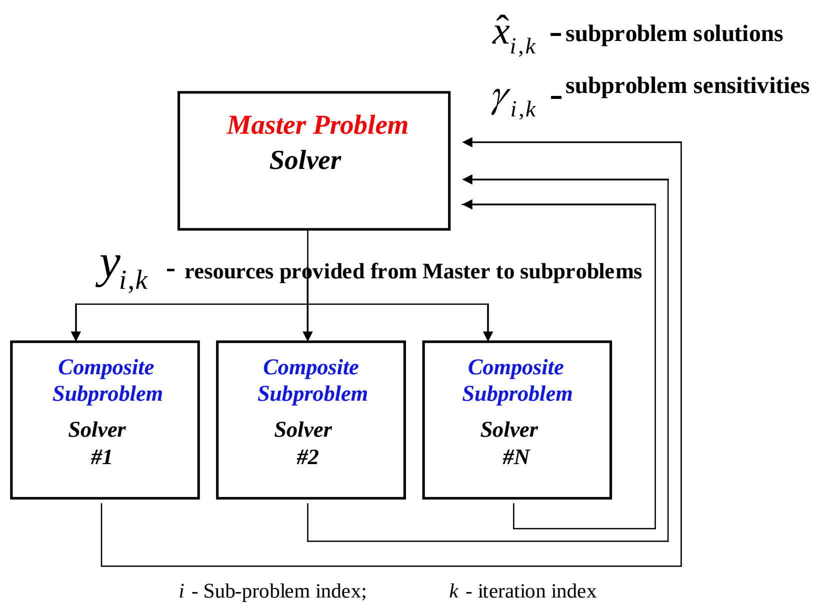
- Solutions to the subproblems and the associated resource sensitivities are sent back to the higher level or *master problem*.
- The available resources are provided from the *master problem* to each of the subproblems.
- If the starting solution is close to optimal, there will be fewer iterations to get to the optimal solution.
- The term MUTEX is used because the master level is providing a constraint-feasible allocation (e.g., MUTEX) of resources.

Figure 4-4: Generic Goal Decomposition

4.5 Iterative Composite Variable Approach

Both goal decomposition and composite variable formulation have their weaknesses. In goal decomposition, if the initial guess at the coordinating variables is far from their optimal values, it may take many iterations before the optimal solution is reached. The solution quality of a problem formulated using composite variables relies heavily on the quality of composite options available to choose from. If poor composites are initially generated, the global solution will suffer as a result. To overcome these limitations, we propose here an approach that combines the strengths of the two techniques into an *Iterative Composite Variable Approach*. In this approach, a pool of composite variables is *iteratively* developed, rather than attempting to pre-calculate a good set of dominating composite variables. In this approach, the decision variables are formulated using composite variables and an iterative approach is used to generate new composites in an "intelligent" way. The "old" composites would be kept as well, resulting in one large pool of composites to choose from. By having a larger pool of composites to choose from, the quality of the solution can at worst stay the same.

Figure 4-5 illustrates an iterative process of arriving at an optimal solution combining goal decomposition and composite variable formulation. Given the value of the decoupling variables $\boldsymbol{y}$ (in our case available resources), each subproblem provides the master level with its best solution defined in terms of decision variables $\boldsymbol{x_i}$ and associated sensitivities $\boldsymbol{\gamma_i}$ [1]. The master problem uses the decision variables and sensitivities to update the values of the decoupling variables (reallocate resources) and improve the overall solution across all subproblems on the next iteration ($\boldsymbol{k}$ iterations). Each of the subproblems can be solved independently. The solution to each subproblem is coordinated with the other subproblems through the decoupling variables.



- Composite Variable Subproblems
  - Optimize their respective subproblem given the available resources $\boldsymbol{y}$ provided to them by the Master Problem
  - Provides solutions $\boldsymbol{x}$ and sensitivities γ back to the Master Problem
- Master Problem
  - Based on the solutions and sensitivities provided by subproblems, reallocates resources to subproblems in effort to generate more valuable overall solution
  - Contains constraint that ensures global solution **does not** use more resources than are available

Figure 4-5: Iterative Solution to a Decomposed Problem

Figure 4-6 [1] focuses the iterative process on the use of composite variables. The process starts with an initial set of subproblem solutions (options). These solutions are also referred to as *plan fragments*. The subproblem solution values $\boldsymbol{x}$ and sensitivities $\boldsymbol{\gamma}$ of the plan fragments are used to create a composite variable formulation which is solved using standard integer programming techniques. This process is repeated at each new iteration until the subproblems have maximized their value subject to resource coordination.

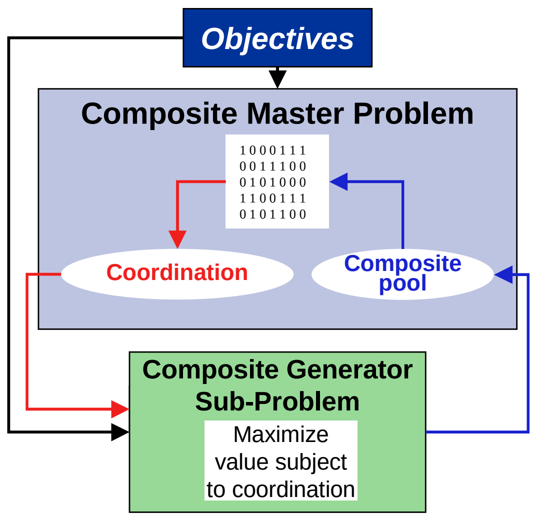


Figure 4-6: Composite Variable Master and Sub-Problem

A specific implementation of the *Iterative Composite Variable Approach* might employ a variety of techniques at each stage of the process outlined above. For instance, the composite variables (plan fragments) might be generated by any number of methods and placed into the composite variable pool. Some of the more common techniques are displayed in Figure 4-7.

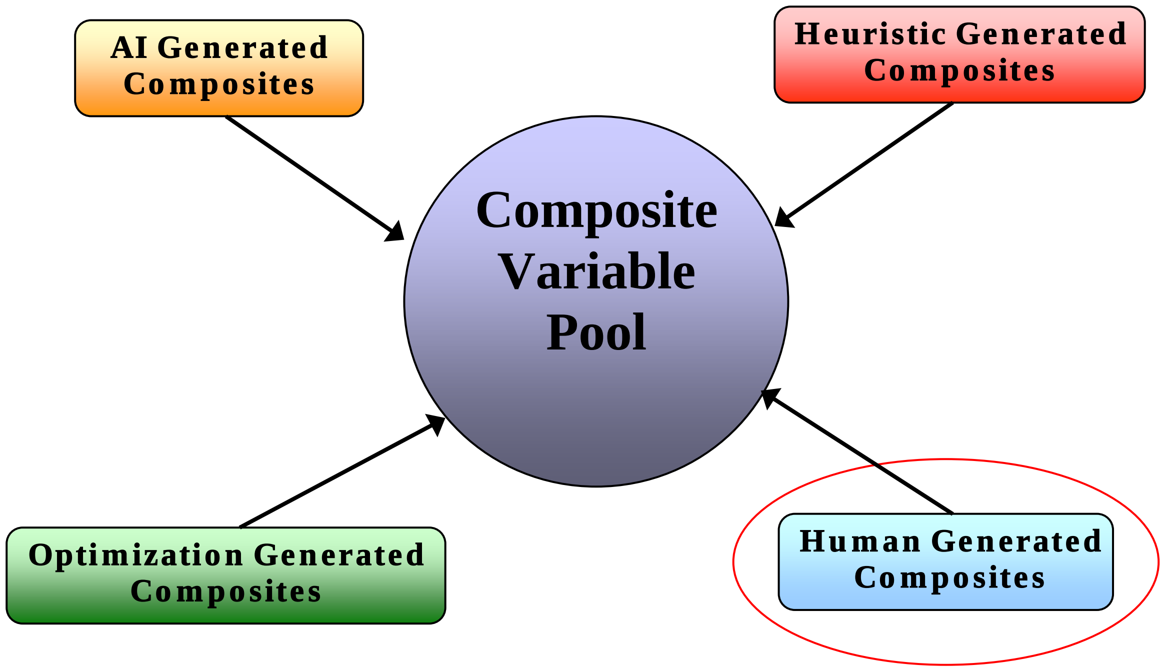


Figure 4-7: Different Techniques for Creating/Updating Composite Variable Pool

Our research focuses on incorporating a human to aid in composite generation. We hypothesize that the use of HMCDM in the creation of the composite variable pool will lead to superior results. Chapter 5 presents experiments whereby human test subjects are permitted to interact iteratively with a computer in an attempt to create an improved pool of composite variables for the MICA problem described in Chapter 3. In these experiments, the subjects create an initial pool of composite variables. Computer algorithms and heuristics are used to generate additional composites which are placed into the pool along side the human generated composites. The subjects are then shown information regarding each of the composites residing within the pool of options. Based on this information, they decide which composites should remain in the composite pool at the next iteration. Their choices are then used to select the plan fragments that are used in the final solution. This *Iterative Composite Variable Approach* takes advantage of the iterative nature of goal decomposition and the combination of decision variables of composite variable formulation.

[This Page Intentionally Left Blank]

Chapter 5

Explanation of HMCDM Experiment

Software developed at The Charles Stark Draper Laboratory designed to simulate a C^2 system of resource allocation and planning entitled Mixed-Initiative Control of Automa-teams (MICA) was used as the simulator for experiments to probe the ideas for human-machine collaboration that have been described in this thesis.

5.1 Experiment Participants

The study consisted of five participants. All participants were employed within the Decision Systems Group of the Control, Information, and Decision Systems Division at Draper Laboratory. All five were graduate students in engineering at the Massachusetts Institute of Technology. In addition, all participants received their undergraduate degrees in Operations Research from The United States Air Force Academy. All of the participants were Second Lieutenants in the Air Force and therefore had knowledge of military planning, objectives, and strategy. None of the participants had previous experience using the MICA platform or any similar resource allocation simulation until these experiments.

The experiments consisted of five independent scenarios, which varied in size and complexity. Size was defined as the number of enemy targets contained within the scenario. Complexity refers to the perceived difficulty. This was determined by the amount of interaction between targets and the amount of overlapping or covering targets. The scenarios were created to gather insight into the benefit of HMCDM with varying levels of size and complexity. In each scenario, the location, type, and damage state of all enemy targets were given with nearly 100% certainty. The details of the scenarios are outlined in Chapter 6. All five participants independently ran each scenario once.

5.2 Goal and Expectations of the Experiment

The objective of the experimental analysis was to determine if incorporating humans into the loop of existing computer optimization planning and resource allocation algorithms would produce better solutions. The experiments were run at three distinct levels of human and machine interaction: human only, human-machine collaboration (HMCDM), and computer only. The hypothesis was that solutions generated through the collaborative effort of a human and a computer would produce better solutions than either human only or computer only solutions. However, this result was not expected in all of the scenarios. For large, very complex scenarios, it was hypothesized that the subjects would have a hard time adding a significant benefit due to trouble grasping all aspects of the entire problem. It was expected that they would be overloaded with too much information and therefore not be able to cluster all of the targets within the scenario efficiently. However, in such scenarios, the subjects might be able to easily identify and efficiently cluster sub-sections of the map allowing them to simplify some portion of the scenario. These same target cluster groupings might not be as "obvious" for a computer, thereby resulting in a significant amount of time and computing power to achieve.

For scenarios small in size, the benefit added by the human was expected to be minimal. In such scenarios, the computer is able to enumerate and evaluate all possible target clustering combinations very quickly and therefore come close to an optimal pool of options in a short amount of time. A significant benefit from human involvement was also expected in scenarios containing numerous enemy targets but not containing complicated threat coverage schemes. In these scenarios, the humans were expected to use spatial reasoning skills and easily identify clustering schemes in a short amount of time. If such scenarios were left solely to the computer, the number of targets would result in considerable computation time and the resulting clusters would most likely be similar to those created by the human operators.

Noteworthy benefits from HMCDM were also expected in medium size scenarios and those scenarios with low to medium complexity. Medium size scenarios are large enough that complete enumeration by the computer in a short amount of time is not realistic. Scenarios with low to medium complexity lend themselves to HMCDM as the key human strength of spatial reasoning and understanding can be combined with the computer visual representation.

5.3 Experimental Simulator

The experimental apparatus consisted of a dual window Graphical User Interface (GUI). This provided the participants with two windows: a left window entitled the "Human System Interface" and a right window entitled "MICA Information Display." Figure 5-1 shows the left window of the dual window display. It contains five main components: a Common Operational Picture (COP) or Map View, information filters, navigation tools, an operator alert button, and team and vehicle information views. The focus in this window was the Map View, which contains a geographic view of target location along with a unique target ID number. In addition, friendly or "blue" assets as well as terrain information like roads, regions of control, etc. are visible in this Map View.

Figure 5-2 shows the right-hand window. It features various information tab panes with controls that present more detail at various levels of the controller hierarchy, a means of navigating between hierarchical levels, a mixed-initiative interaction panel for accepting or rejecting controller requests, outstanding decision and mission log tables, and a message log window.

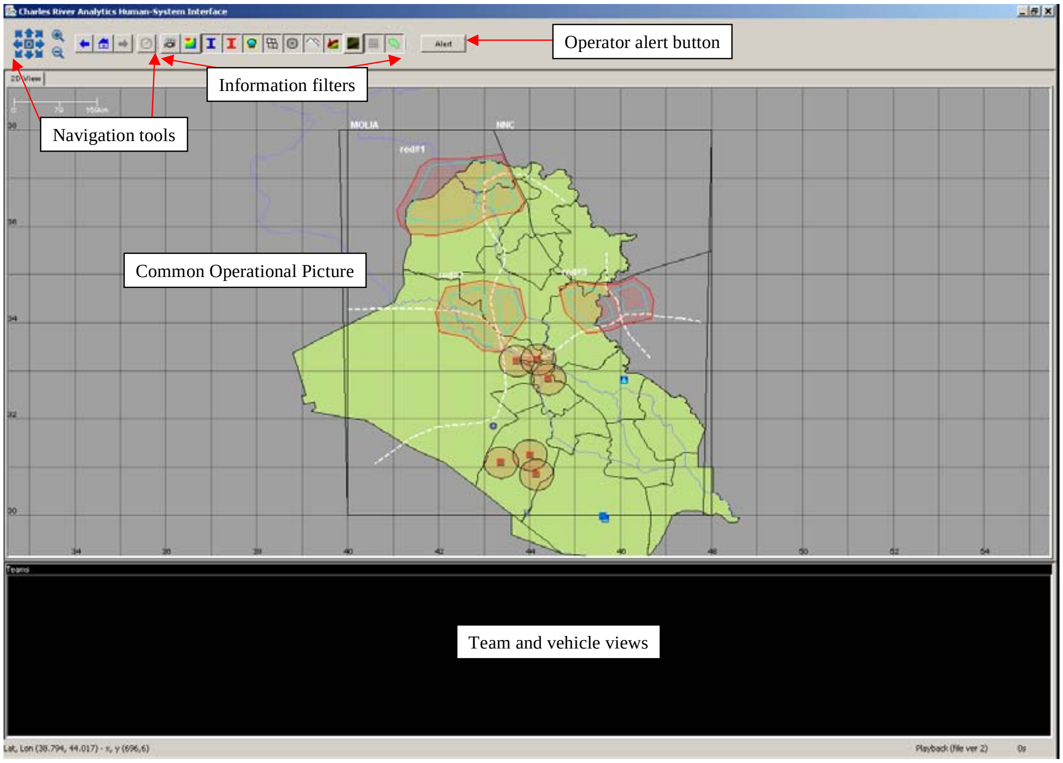


Figure 5-1: Human System Interface (Left-Hand Display)

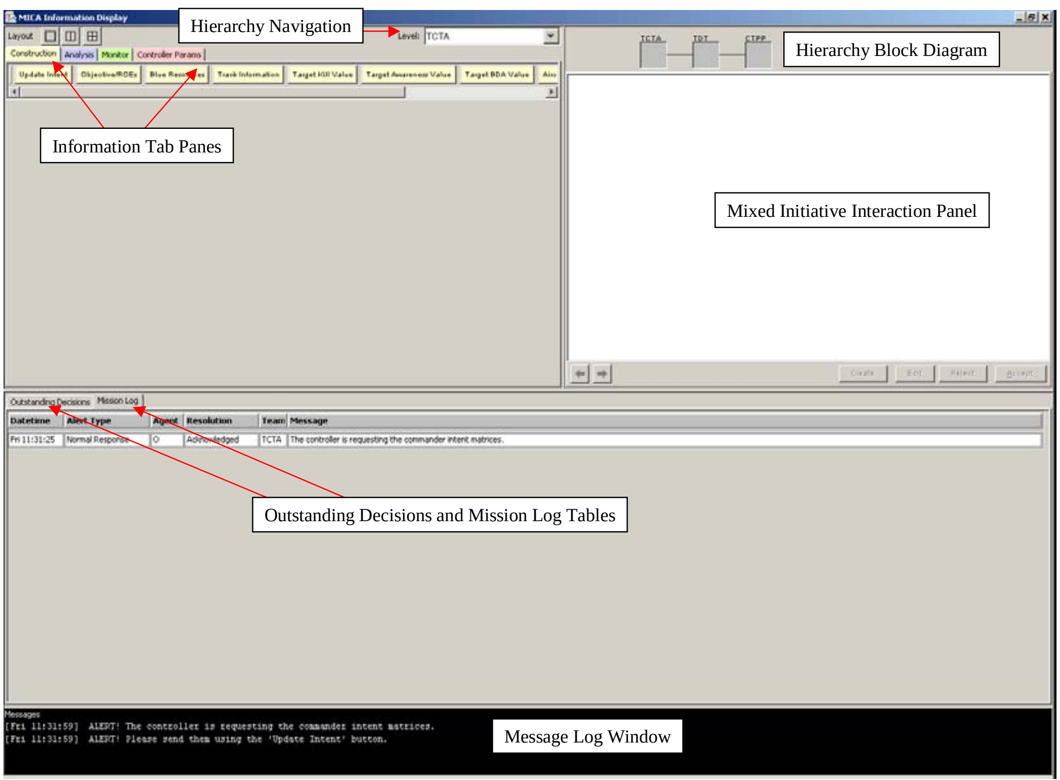


Figure 5-2: MICA Information Display (Right Hand Display)

Not all of the available features in the MICA Information Display were used in this particular experiment. However, the features that were used are outlined throughout this chapter.

Figure 5-3 shows the top-level objectives and Rules of Engagement (ROEs) display. It gives a synopsis of the commander's objectives for this mission as well as what ROEs must be followed for mission success.

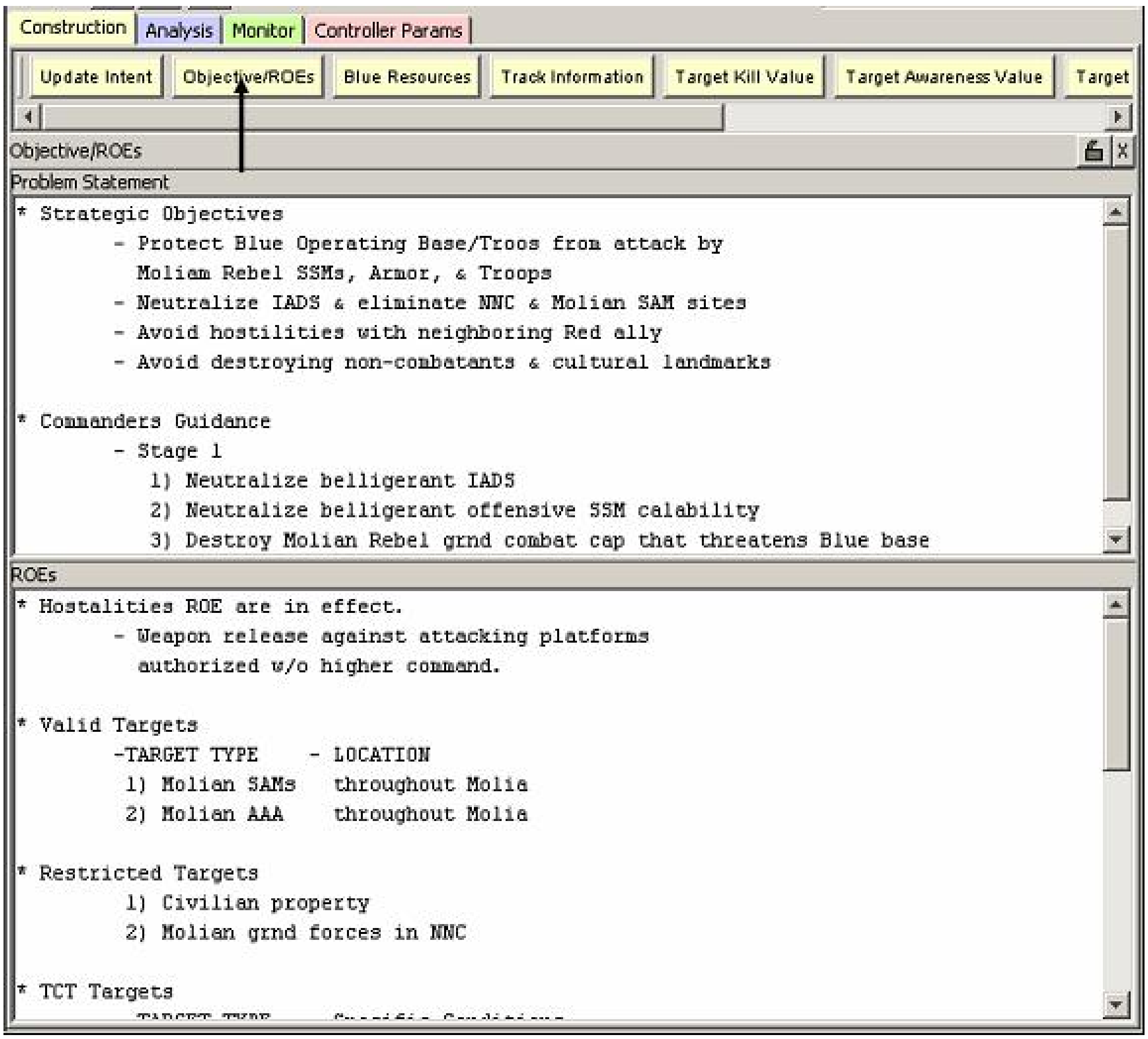


Figure 5-3: The Top-Level Objectives and ROE Component

From the "Blue Resources" menu tab, Figure 5-4, the users were able to obtain information on which aircraft were available for each of the given scenarios.

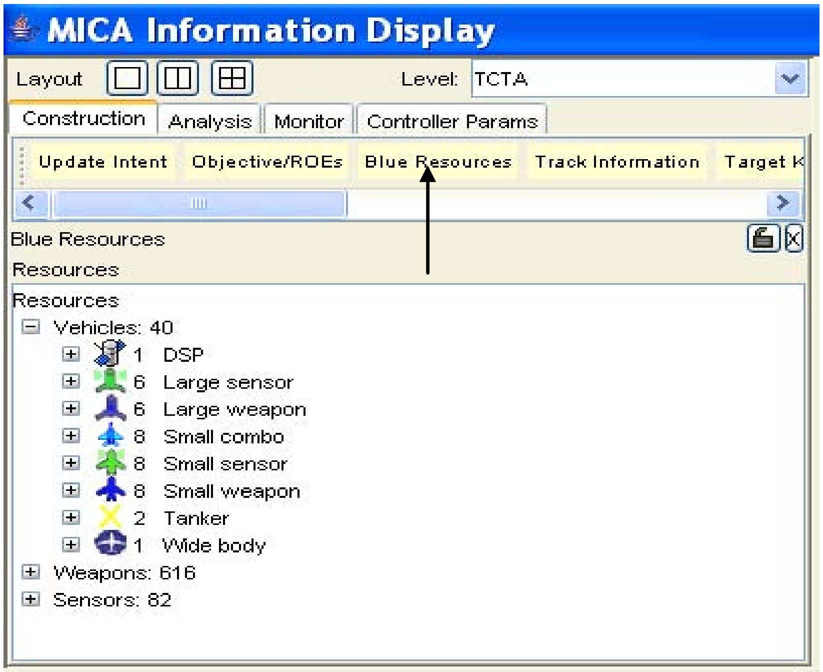


Figure 5-4: Blue Resources Menu

The "Track Information" menu, Figure 5-5, lists all known tracks. It breaks out the tracks into identified and unidentified where the unidentified are any track whose type certainty is less than 50%. The ID, type certainty, damage certainty, and destruction certainty for each track is listed as well.

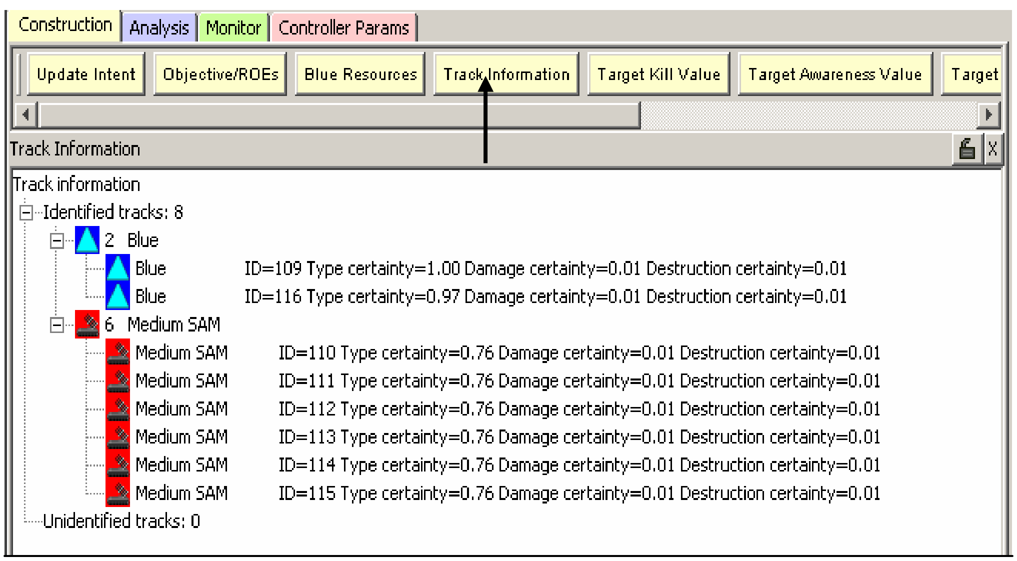


Figure 5-5: Track Information Menu

The “Target Kill Value” menu tab, Figure 5-6, shows what value is associated with destroying each one of the target types within the scenario. The MICA Information Display also contains the menu tabs “Target Awareness Value”, “Target BDA value”, and “Aircraft Value.” Each of these menus is very similar to the “Target Kill Value” menu shown below in Figure 5-6. In addition to standardizing values for targets of similar types, it is also possible to adjust the value of specific enemy tracks of the same type. This is done in the “Per Track Importance” menu tab (Figure 5-7). Within the MICA framework, all values contained in each of these menu tabs are adjustable. However, for the purposes of this experiment, the values were set at default values and the subjects were not able to adjust them.

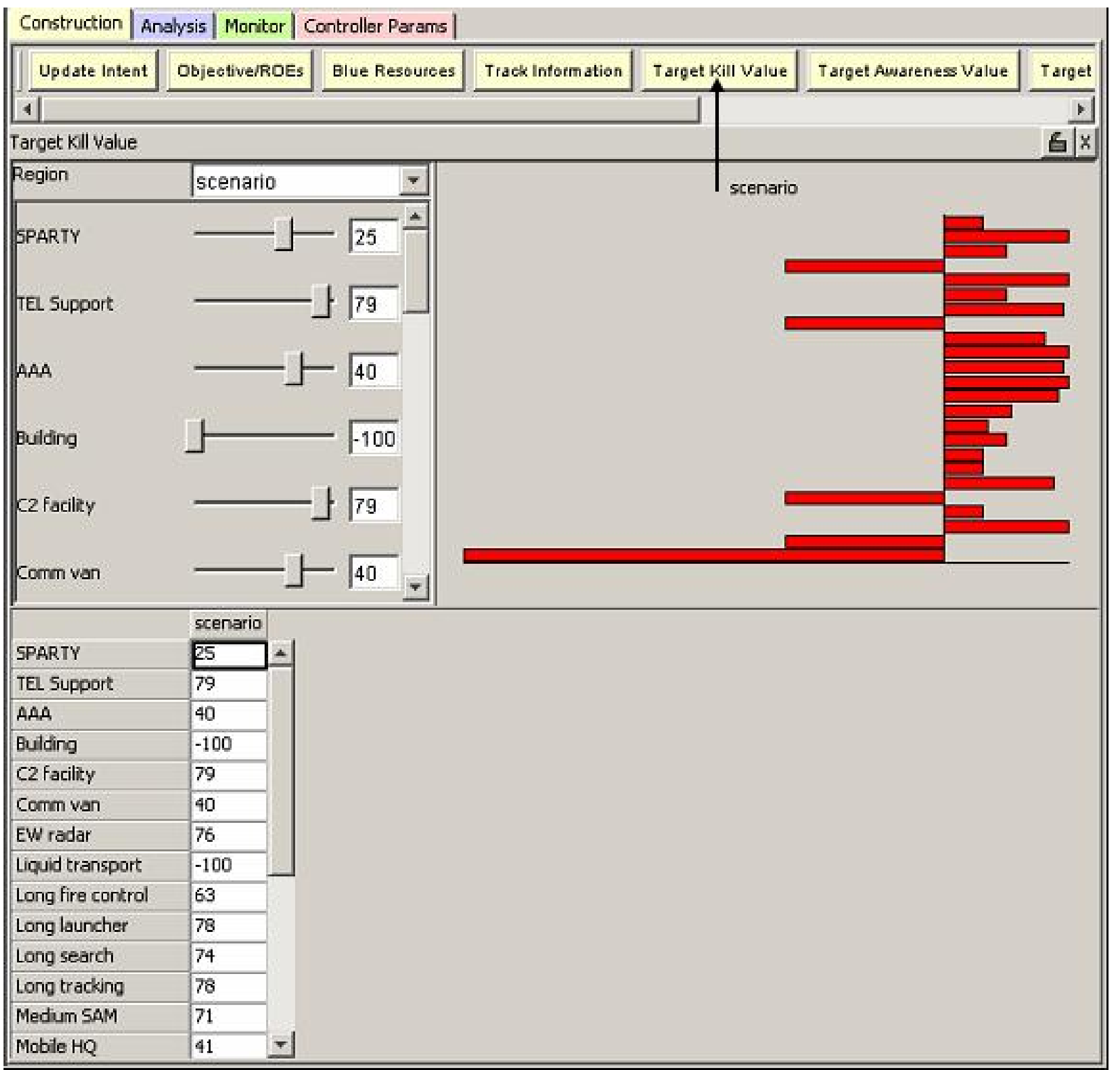


Figure 5-6: Enemy Target Type Value

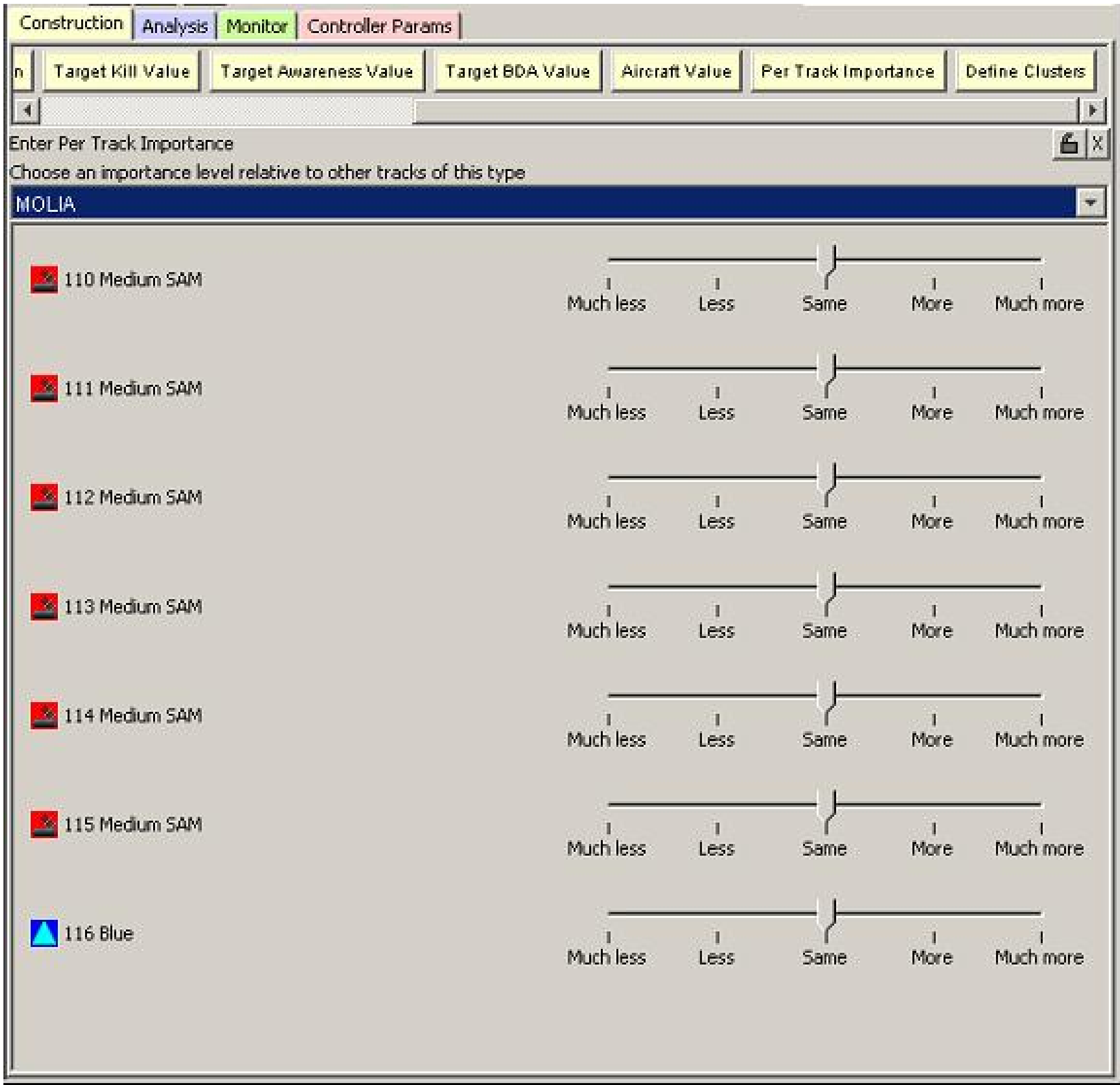


Figure 5-7: The Per Track Importance Component

The friendly assets in these scenarios consisted of several types of unmanned aerial vehicles (UAVs). There are five primary types of UAVs: *large sensor*, *small sensor*, *large weapon*, *small weapon*, and *small combo*. Each aircraft platform type has different possible configurations of sensors and weapons. For example, "weapon" aircraft can only carry weapons, "sensor" aircraft can only carry sensors, and "combo" aircraft can carry both weapons and sensors. The adjectives small and large refer to how many weapons or sensors the aircraft can carry. For example, a *large weapon* aircraft can carry twenty weapons while a *small weapon* aircraft can only carry eight. The UAVs that were capable of carrying weapons were restricted to use a single weapon type, the homing missile. The homing missiles can be delivered to any target within a 40 km radius. There were three main types of enemy targets common to all five scenarios: the Long Launcher, the Medium Surface to Air Missile (SAM) sites, and the TEL Support. The Long Launcher is capable of firing surface-to-air missiles at UAVs a maximum of 80 km away, significantly outranging the homing missile. This imbalance requires that a UAV safely navigate deep within a Long Launcher's engagement zone in order to attack it. The use of two UAVs is needed in order to safely perform such a maneuver; one to jam the Long Launcher's ability to track aircraft and the other to fly inside the threat zone to a distance within 40km to attack the target. The default value for destroying a Long Launcher is 15. The Medium SAM can only fire at targets within a distance of 38 km. This makes it possible for a single UAV to safely attack a lone Medium SAM. It is not necessary to apply the standoff-jamming tactic against a Medium SAM as long as the location of the SAM is sufficiently known. The default value for destroying a Medium SAM is 25. The TEL Support does not have the ability to fire weapons. Therefore, it does not pose a significant threat to any of the unmanned vehicles. However, these targets are given a high priority, they are worth a value of 300 if destroyed.

5.4 *Procedures*

5.4.1 Training Scenario

At the beginning of the experiment, the overarching goals of the scenarios and the procedures were described to each of the participants. They were also told how they would be evaluated at the conclusion of their experiment. They were told to cluster enemy targets in a way that would generate the maximum overall solution value while trying to destroy high value targets as quickly as possible, using the minimum number of resources possible (both weapons and aircraft), with minimum aircraft attrition, while conducting the entire mission in the shortest amount of time possible. They were also told that they would be timed throughout the simulation. More will be discussed on the evaluation metrics in Section 5.5.

The participants then received training, which consisted of observing an example scenario and performing dry runs. In the observed example, a human operator made the decisions of which targets to cluster and which target-aircraft options to choose. Participants were given unlimited time with the simulator to conduct as many dry runs as necessary to become familiar with the software. The practice runs were performed on one, basic scenario, shown in Figure 5-8.

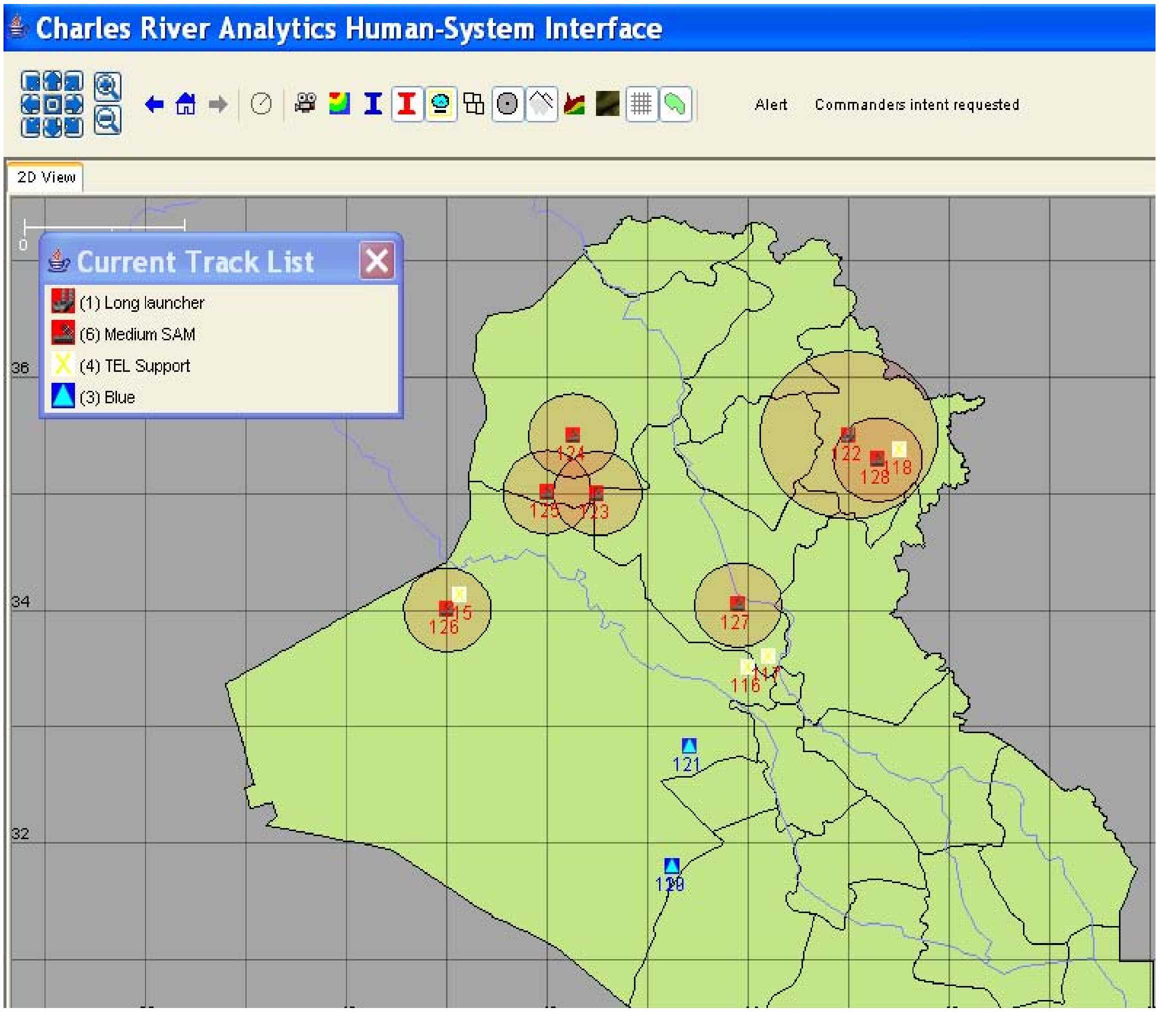


Figure 5-8: Training Scenario

The training scenario, shown in Figure 5-8, was created to allow the subjects to become familiar with different target types as well as different threat coverages. As seen in Figure 5-8, Targets 123, 124, 125, 126, 127, & 128 are all Medium SAM sites; Targets 115, 116, 117, & 118 are TEL Support's; and Target 122 is a Long Launcher. The scenario was also designed to allow the participants to become familiar with the different types of threat coverage, which are illustrated by the size of each target's associated threat ring. The three types of coverage used in this simulation are full, multiple, and no coverage. The training scenario was designed to include at least one example of each of these types of coverage. Full coverage does not allow the participants to simply attack the TEL Support Target 115 by itself, in the training scenario, because the Medium SAM Target 125 would have to be destroyed first. Multiple coverage is depicted in the training scenario using Targets 118, 128, & 122. The TEL Support Target 118 cannot be destroyed until the Medium SAM Target 128 is destroyed which itself cannot be attacked until the Long Launcher 122 is taken out. Finally, targets 116, 117, 123, 124, 125, & 127 could each be attacked independently without having to destroy any other targets first, which represents no coverage. These examples are present in the basic scenario so the users can develop their strategies during the training scenario in order to reduce the learning effect that would otherwise occur between the scenarios. Each participant was given as much time as they wanted to familiarize themselves with the program.

5.4.2 Initial Human Cluster Selection

Once the subjects were satisfied that they no longer needed practice on the training scenario, the experiment scenarios commenced. The information available in the menus described in Section 5.3 was used by the subjects to select and submit clusters of targets to the computer for initial evaluation. This was done through the use of a text file in which the users select the target IDs they wished to be clustered together. Each line in the text file signified a separate cluster of targets. For example, Figure 5-9 is an illustration of possible clusters the user might have entered based on the training scenario in Figure 5-8. This input corresponds to four distinct enemy clusters: one cluster includes Targets 115 & 126, one contains Targets 116, 117, & 127; and so on. It is important to note that the ordering of the Target ID numbers within each cluster (i.e. Target 115 being placed "before" 126 in the first cluster) as well as the ordering of the clusters on the respective lines in the text file (i.e. the cluster with 115 and 126 being on the first line of the text file) were not significant. The information was read into the system in the same way even if the user would have entered the third line as {124 123 125} or if the clusters on the fourth line and first line were switched.

Once the user decides which targets they would liked clustered, they input them into the text file as noted above, save the text file, and then click the "Commander's Intent" button in the MICA Information Display GUI to have this information sent into the program, Figure 5-9 and Figure 5-10.

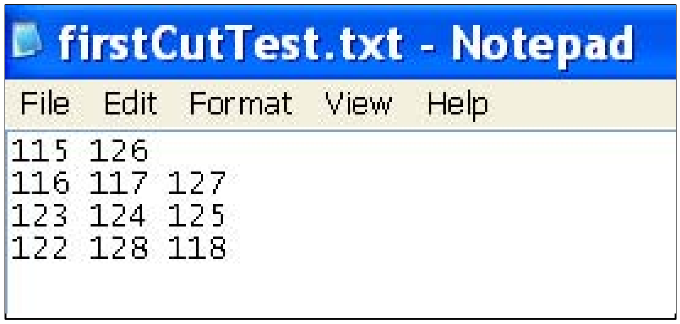


Figure 5-9: Human User Inputting Enemy Target Clusters for Evaluation

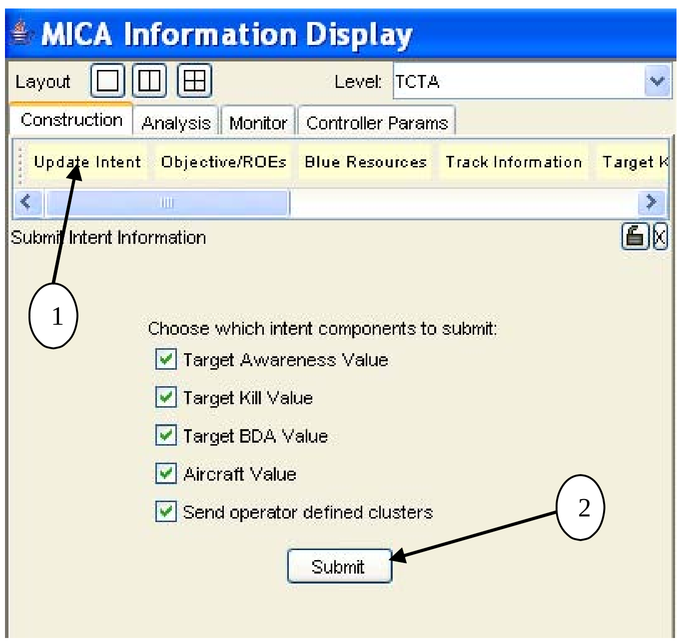


Figure 5-10: Sequence for User to Send in Initial Human Generated Clusters

The computer takes this initial set of human generated clusters and begins to build upon each of the composite variables by adding on teams of aircraft to clusters of enemy targets. Each of these enemy cluster-aircraft team pairings is referred to as a *cluster-team option***.** In some cases, the computer will add multiple different teams of aircraft to the same cluster, with each different team-cluster corresponding to a unique option. For example, a cluster containing Targets 116 & 117 and assigned a team consisting of one small combo aircraft and one large weapon aircraft *is a different option* from one that the contains the same Targets 116 & 117 but has assigned a different team of aircraft (i.e., one small combo and one small sensor). In addition to the computer adding aircraft teams to the human specified clusters, the computer will also generate entire options on its own. The computer follows the same process as the human operator by first choosing enemy targets to cluster together and then assigning aircraft to these clusters. Further description on this process is provided in Chapter 3.

5.4.3 "Human Only" Experiments

There were two separate types of experiments that involved a human, a "Human Only" experiment and a "Human-Machine Collaboration" experiment. The terms referred to the amount of human involvement in the initial creation of the composite variables which in this case is the clustering of enemy targets. The procedures varied slightly for each. For the creation of clusters by the human only, the following set of procedures was used:

Human Only Creation of Clusters

- Human looks at map generated by MICA showing location and type of enemy targets.
- Human inputs finalized clusters based solely upon their inspection of the map layout.
- Computer adds on details to these clusters at lower levels of planning.
- Metrics are obtained at the conclusion of the experiment for analysis into the quality of solution.

Table 5-1: Flow of Experiment for Human Only Creation of Clusters

5.4.4 "Human-Machine Collaboration" Experiments

The difference in the HMCDM experiments was the fact that the involvement represented an iterative process. This iterative process was created based on the *Iterative Composite*

Variable Approach represented in Chapter 4. In these experiments, the humans entered an initial set of clusters which were evaluated by a computer. The results of this evaluation were displayed back to the humans in the form of the Key Pieces of Information (KPI). Based on this information, the humans were then able to modify or select clusters (composites contained in the composite variable pool) in an attempt to achieve the global optimal solution to the scenario.

Human-Machine Collaborative Cluster Creation

- Human looks at map generated by MICA showing location and type of enemy targets.
- Human inputs clusters they wish to gain more information about into a text file.
- Input file with human clusters read into the MICA system.
- Computer adds on aircraft teams to human created clusters, these are now referred to as *cluster-team options.*
- Key Pieces of Information (KPI) about each of these options is created.
- Computer generates own cluster-team options using existing system heuristics.
- KPI about computer generated options are gathered.
- KPI for all options (both human and computer generated) are output to the human user.
- Human uses all of the given KPI in order to decide which options he/she wants to select for the final solution.
- User inputs their desired options for the final solution into a second text file.
- Computer checks for overlapping resources in chosen options.
    - If aircraft resources overlap, computer checks for availability of same type of aircraft to use instead.
        - If same type of aircraft available, computer substitutes this aircraft into one of the clusters.
        - Otherwise, computer will de-conflict the overlapping options by selecting the option(s) that fit into the overall plan the best.
    - If enemy targets overlap, computer selects option(s) which fit into overall plan the best.
- Metrics are obtained at the conclusion of the experiment for analysis into the quality of solution.

Table 5-2: Flow of Experiment for Human-Machine Collaborative Creation of Clusters

5.4.5 Key Pieces of Information (KPI)

The KPI are the pieces of information the human operator needs in order to make intelligent choices as to which cluster-team pairing options should be chosen in order to create the best global solution. Methods for determining which pieces of information should be contained in the KPI are qualitative because they are very dependent on the context. In each particular program or system, the KPI could be completely different; however, the main goal of the KPI remains the same. This goal is to strike the appropriate balance between ensuring the human is given enough information to make intelligent choices yet not overwhelming them with too much information.

The KPI chosen for this research are listed below:

- Targets - The targets that are contained within each option. In a human generated cluster, these targets are specified by the human operator whereas in the computer generated clusters, the targets are selected by the heuristics in the automation.

- Aircraft - The aircraft assigned in each option. The aircraft teams assigned to each cluster of enemy targets are always done by the computer in this experiment. However, future research could examine if there is a benefit to allowing the human operator the option of controlling this function and specifying the aircraft teams themselves. This idea was discussed in Section 3.3.2.

- Value – The value associated with each cluster-team option. This information is obtained from lower levels in the MICA planning hierarchy. After the cluster-team options are sent down the hierarchy, the lower planning levels decide what actions should be taken on each of the enemy targets within the cluster with respect to which aircraft are assigned to the cluster. Based on these decided actions (sensing the target, attacking the target, any combination of sensing and attacking, or carrying out no action), the system is able to calculate the value that is generated based on the cluster-team options chosen.

- Time – The time to carry out the actions within each cluster-team option. This information is also gathered from lower levels of the MICA planning hierarchy.

Once the lower levels decide what actions should be taken on each of the targets they then calculate the expected time to carry out these actions.

- Risk – Risk is calculated by summing the probability of attrition for all of the aircraft within each cluster-team option. Again, this information is gained from the lower levels in the MICA planning hierarchy. After the automation decides what actions are to be taken on each of the enemy targets, the lower planning levels determine which aircraft will carry out these actions, the specific flight routes to achieve these actions, and the geographic points on the map from which the aircraft will sense or attack. Based on this information, the system is able to calculate the risk associated with each cluster-team option.

- Efficiency of Resources – The 'Efficiency of Resources' piece of information pertains to both the weapons and sensors that are planned for use by the aircraft in each cluster-team option. One of the goals is not to waste resources. If there is a cluster consisting of only one or two enemy targets, only one, or at the most two aircraft are needed to meet the mission goals. The efficiency is measured in terms of percent usage. For example, if 40 weapons are available due to the aircraft in the option but the plan includes using only 4 of those weapons, then only 10% of the weapon resources are used.

- Cluster created by whom – Each option is also labeled as to whether the human created the enemy cluster or whether it was entirely computer generated. We felt this would be an interesting aspect to study, capturing whether the human user had a preference for ultimately choosing their own options or if they felt more confident in the computer generated options.

The KPI are output to the human via a text file. Figure 5-11 is an example of these KPI. Each option is numbered starting with OPTION 0. The KPI is presented to provide the human with the information necessary to evaluate each of the given options and choose the subset of options that will lead to the best overall global solution. Again, there is no single metric to tell the user which option is the 'best' or which combinations of options are the ‘best’. They are to pick the options based on the top-level objectives, ROEs, (Figure 5-3) and the following evaluation metrics: overall solution value, time to destroy high value targets, number of resources used (both weapons and aircraft), aircraft attrition, the amount of time the entire mission took in the simulation, and the amount of real world time for the mission to conclude. These metrics are described in further detail in Section 5.5.

Option ID	Generated	Targets	Aircraft	Weapon Efficiency	Value	Cost	P(Attrition)	Mission Time
0	HUMAN	126 129 146 124 145 125	large_weapon_type 2; small_combo_type 3;	12.0 / 24.0	20	30.00022	0.999996755	4h 27m 26s
1	HUMAN	126 129 124 146 125	large_weapon_type 2; small_combo_type 3;	10.0 / 24.0	734	3.09E-04	0	4h 17m 23s
2	HUMAN	126 146 124 145 125	large_weapon_type 2; small_combo_type 3;	10.0 / 24.0	20	30.00021	0.999996755	4h 19m 12s
3	HUMAN	126 129 146 125	large_weapon_type 2; small_combo_type 3;	8.0 / 24.0	502	2.81E-04	0	3h 54m 10s
4	COMPUTE	126 129 146 133 125 130	large_weapon_type 2; small_combo_type 3;	12.0 / 24.0	543	3.72E-04	0	5h 10m 8s
5	COMPUTE	124 143 145 144	large_weapon_type 2; small_combo_type 3;	8.0 / 24.0	291	2.95E-04	0	4h 6m 10s
6	COMPUTE	134 136 135	large_weapon_type 2; small_combo_type 3;	6.0 / 24.0	60	2.63E-04	0	3h 39m 23s
7	COMPUTE	141 140 142	large_weapon_type 2; small_combo_type 3;	6.0 / 24.0	60	2.02E-04	0	2h 47m 59s
8	COMPUTE	137 138	large_weapon_type 2; small_combo_type 3;	4.0 / 24.0	40	3.05E-04	0	4h 14m 14s
9	COMPUTE	128 139	small_combo_type 3; large_weapon_type 2;	4.0 / 24.0	40	2.35E-04	0	3h 15m 45s
10	COMPUTE	131 132	small_combo_type 3; small_combo_type 3;	4.0 / 8.0	40	20.00008	2	1h 9m 51s
11	COMPUTE	128 139	small_combo_type 3; small_combo_type 3;	4.0 / 8.0	40	9.86E-05	0	1h 22m 11s
12	COMPUTE	137 138	small_combo_type 3; small_combo_type 3;	4.0 / 8.0	40	1.24E-04	0	1h 43m 23s

Figure 5-11: Key Pieces of Information (KPI) shown to human user

Once the human decides which options to include in the final solution, they input their choices again through a text file. For example, if they wished to include Options 0, 1, and 4; they enter the numbers 0, 1, and 4 on the first line of the text file, resave the file, and then press any key for the program to start running again (see Figure 5-12).

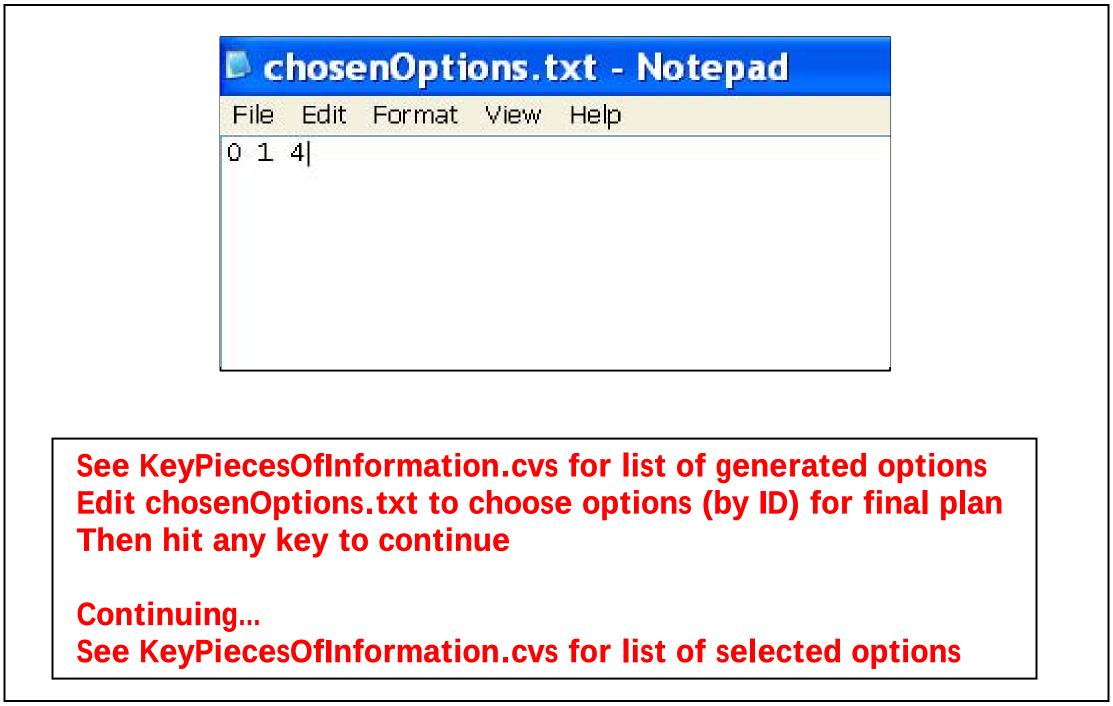
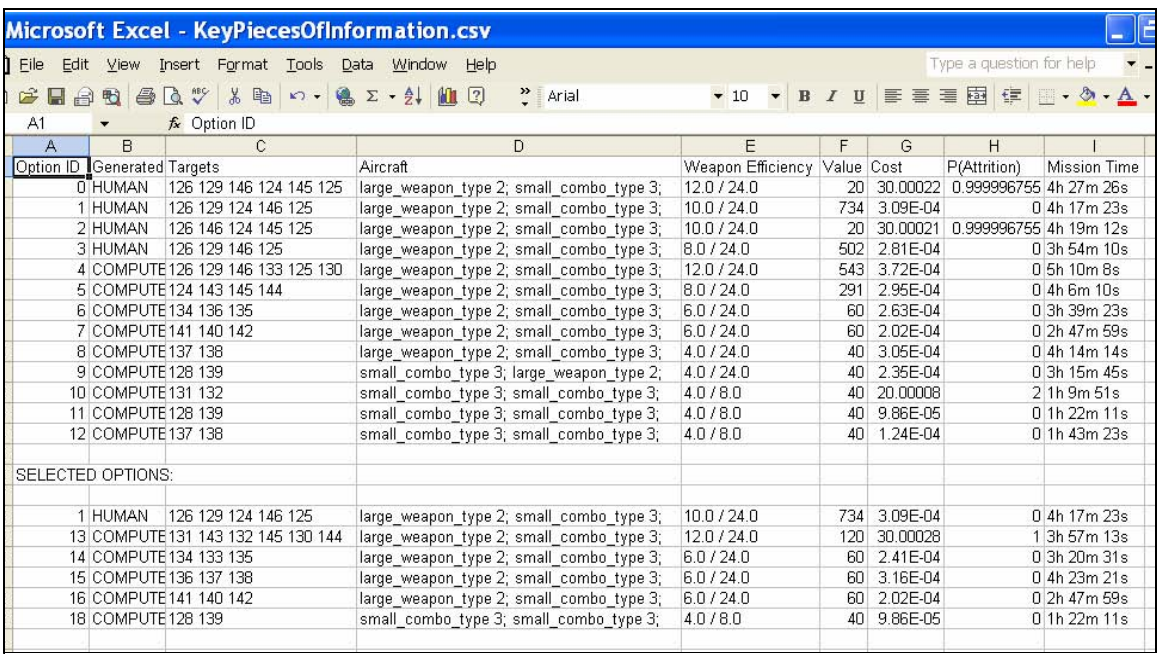


Figure 5-12: Human Choosing Cluster-Team Options



Option ID	Generated	Targets	Aircraft	Weapon Efficiency	Value	Cost	P(Attrition)	Mission Time
0	HUMAN	126 129 146 124 145 125	large_weapon_type 2; small_combo_type 3;	12.0 / 24.0	20	30.00022	0.999996755	4h 27m 26s
1	HUMAN	126 129 124 146 125	large_weapon_type 2; small_combo_type 3;	10.0 / 24.0	734	3.09E-04	0	4h 17m 23s
2	HUMAN	126 146 124 145 125	large_weapon_type 2; small_combo_type 3;	10.0 / 24.0	20	30.00021	0.999996755	4h 19m 12s
3	HUMAN	126 129 146 125	large_weapon_type 2; small_combo_type 3;	8.0 / 24.0	502	2.81E-04	0	3h 54m 10s
4	COMPUTE	126 129 146 133 125 130	large_weapon_type 2; small_combo_type 3;	12.0 / 24.0	543	3.72E-04	0	5h 10m 8s
5	COMPUTE	124 143 145 144	large_weapon_type 2; small_combo_type 3;	8.0 / 24.0	291	2.95E-04	0	4h 6m 10s
6	COMPUTE	134 136 135	large_weapon_type 2; small_combo_type 3;	6.0 / 24.0	60	2.63E-04	0	3h 39m 23s
7	COMPUTE	141 140 142	large_weapon_type 2; small_combo_type 3;	6.0 / 24.0	60	2.02E-04	0	2h 47m 59s
8	COMPUTE	137 138	large_weapon_type 2; small_combo_type 3;	4.0 / 24.0	40	3.05E-04	0	4h 14m 14s
9	COMPUTE	128 139	small_combo_type 3; large_weapon_type 2;	4.0 / 24.0	40	2.35E-04	0	3h 15m 45s
10	COMPUTE	131 132	small_combo_type 3; small_combo_type 3;	4.0 / 8.0	40	20.00008	2	1h 9m 51s
11	COMPUTE	128 139	small_combo_type 3; small_combo_type 3;	4.0 / 8.0	40	9.86E-05	0	1h 22m 11s
12	COMPUTE	137 138	small_combo_type 3; small_combo_type 3;	4.0 / 8.0	40	1.24E-04	0	1h 43m 23s
SELECTED OPTIONS:								
1	HUMAN	126 129 124 146 125	large_weapon_type 2; small_combo_type 3;	10.0 / 24.0	734	3.09E-04	0	4h 17m 23s
13	COMPUTE	131 143 132 145 130 144	large_weapon_type 2; small_combo_type 3;	12.0 / 24.0	120	30.00028	1	3h 57m 13s
14	COMPUTE	134 133 135	large_weapon_type 2; small_combo_type 3;	6.0 / 24.0	60	2.41E-04	0	3h 20m 31s
15	COMPUTE	136 137 138	large_weapon_type 2; small_combo_type 3;	6.0 / 24.0	60	3.16E-04	0	4h 23m 21s
16	COMPUTE	141 140 142	large_weapon_type 2; small_combo_type 3;	6.0 / 24.0	60	2.02E-04	0	2h 47m 59s
18	COMPUTE	128 139	small_combo_type 3; small_combo_type 3;	4.0 / 8.0	40	9.86E-05	0	1h 22m 11s

Figure 5-13: Representation of Human Selected Options

The options that the human selects will be included in the solution as long as the options represent a feasible set of solutions. The only case in which not all of the human chosen options are included in the final solution is the case of the human choosing different options that contained the same targets.

It is important to note that the collection of all cluster-team options do not represent a set of mutually exclusive elements. For example, there might be multiple options that contain the same targets or the same aircraft. The number of times the subjects chose options that contained overlapping targets or aircraft was recorded. This was done to see how many options became *too many* options. It was hypothesized that if there were an excessive number of options for the human to choose from, they might become overwhelmed and might mistakenly choose options that contain the same targets or the same aircraft.

A hypothesis was also made that in some cases, the user might actually want to choose overlapping options. If there were multiple options that the user felt were equally as good yet they contained overlapping items, they might be indifferent as to which option they select, as long as at least one of them is included in the final solution. In order to resolve this, after the experiment, the subjects were asked if they selected options with overlapping items on purpose or by accident due to being overwhelmed or confused. In the event that the human selected multiple options with overlapping targets, the system ran through its algorithm of evaluation and selected which options fit more easily into the overall plan. For example, the algorithm would remove the lowest valued overlapping option until a feasible solution was reached. However, the computer's process for handling options with overlapping aircraft was slightly different. If the human selected options with overlapping aircraft, the program would first search the inventory of available aircraft to see if there was another aircraft of the exact same type that was not being used. If there were not any available aircraft of the same type, then the computer would choose whichever aircraft option fit more easily into the overall plan.

After the participant input their desired clusters and resumed the program, the initial file in which the options and KPI were listed was appended at the end to include which options were ultimately selected to be part of the overall solution.

From this point on, the control was left entirely to the computer. After the system was given the selected options, this information was sent to the lower levels of the MICA hierarchical program to add the remaining pieces of information to finish creating each composite variable. In this particular case, the human supplied the computer with the enemy targets to be contained in each cluster along with the associated aircraft to be assigned to each cluster. The computer then added on information as to which aircraft would perform each particular action against each enemy target. The lower levels of the MICA planning hierarchy also added on routing information for each of the aircraft, locations on the map for aircraft to release weapons and/or use sensors, etc. Refer back to Chapter 3 for further information on the hierarchical creation of the composite variables.

5.5 Evaluation of Human Involvement

As it was mentioned in the beginning of this chapter, the goal of this experiment was to gain insight into the optimal way to incorporate the strengths of a human operator with the strengths of a computer to create the "best" possible solutions. Therefore, in order to evaluate human involvement, metrics were identified to determine if adding the human-in-the-loop for the creation of target clusters added any benefit to the overall solution. In addition to each of the five participants running the five scenarios, each scenario was also run without any human involvement. These solutions were used as our baseline for comparison and were referred to as "computer only" solutions. Because of the complexity of the problem of C2 resource allocation and mission planning, there are many metrics that determine the quality of a solution. For this experiment a qualitative evaluation, which takes into account multiple quantitative metrics, was used. In military mission planning there is an inherent trade-off between risk and return. Also, the return is not always obvious. For example, in some situations it is necessary to develop a plan in the shortest amount of time possible, despite the quality of the plan. At other times, the number of resources used might be the most crucial return indicator. Therefore, metrics for risk and return, on multiple levels were used to evaluate the quality of the solutions for each scenario. The details of the metrics chosen are described below:

- Value of the overall solution – The value of the overall solution was obtained by adding the value obtained from each of the cluster-team pairings throughout the life of the scenario. The beginning of Chapter 3 provides a further description of the value. This value was described in more detail in the beginning Chapter 3.

- Time to create the initial mission plan – This metric is important because although there may be plan solutions that contain more value, they might also take significantly more time to create. Analysis of this data will allow for comparison of the overall solution vs. time trade-off.

- Attrition of aircraft – This is an essential metric because there is a possibility that solutions might be generated with higher value and/or solutions that have the mission conducted faster but that also might have a higher attrition of aircraft.

- Number of weapons used – If there are two equivalent solutions yet one is using considerably less weapons to achieve the same solution, this will be the superior option. The only weapons available for these experiments were homing missiles.

- Number of aircraft used – All else being equal, it is best to implement a solution that uses fewer aircraft.

The degree of overlap in the solutions created by a human user versus those generated solely by the computer was another interesting piece of information examined at the end of the experiment. It was analyzed to give insight into the similarities of the human and computer strategies. If there was a large amount of overlap in multiple scenarios this would show that there was not much gained by placing a human-in-the-loop.

Finally, information was gathered on how often the human users chose their own created clusters versus how often they chose clusters that were created by the computer. This data allows conclusions to be drawn on the human's confidence in their own solutions versus computer created solutions.

[This Page Intentionally Left Blank]

Chapter 6

Results of HMCDM Experiments

This chapter describes the HMCDM experiment scenarios and results in detail. Five scenarios were run at three distinct levels of automation with respect to the involvement in the process of clustering enemy targets: human only, human-machine collaboration, and computer only. The chapter begins with highlighted results from the experiments. Analysis of the individual scenario results as well as broad analysis encompassing the collection of scenarios is given later in the chapter. All averages discussed for the human only and human-machine collaboration experiments were calculated as an average of the results from each of the five participants. There was only one result for the computer only experiments, therefore an average was not necessary. The computer only results were generated on a run of the experiment without collaboration for each scenario. The value of the solution equals a summation of the value accumulated from each target "hit" within the scenario. Determination of individual target values is described in detail in Chapter 3.

6.1 Overall Results

Figure 6-1, Figure 6-2, and Figure 6-3 provide comparisons of the solution value generated using each level of automation. Figure 6-1 breaks down the average value generated in each scenario for the varying levels of automation. This figure depicts that with some level of human involvement, equal or higher valued solutions were created in every scenario. Figure 6-2 further illustrates the results for the average value achieved in each scenario by the human only, human-machine collaboration, and machine only levels of automation. Standard error bars are given for the human only and human-machine collaboration results. Figure 6-3 provides a normalized comparison of the solution values. This figure shows that on average the human-machine collaborative plans produced the highest value.

Summary Results
Avg Scenario Value Generated

	Human Only Clustering	Human-Machine Collaboration	Computer Only Clustering
Wall Scenario	555	830	25
Ring Scenario	565	555	565
Small Complex Scenario	64	52	0
Large Simple Scenario	842	1130	959
Challenge Problem Scenario	207	170.4	115
Mean	**446.6**	**547.48**	**332.8**

Figure 6-1: The Average Value Generated for each Scenario and Level of Automation

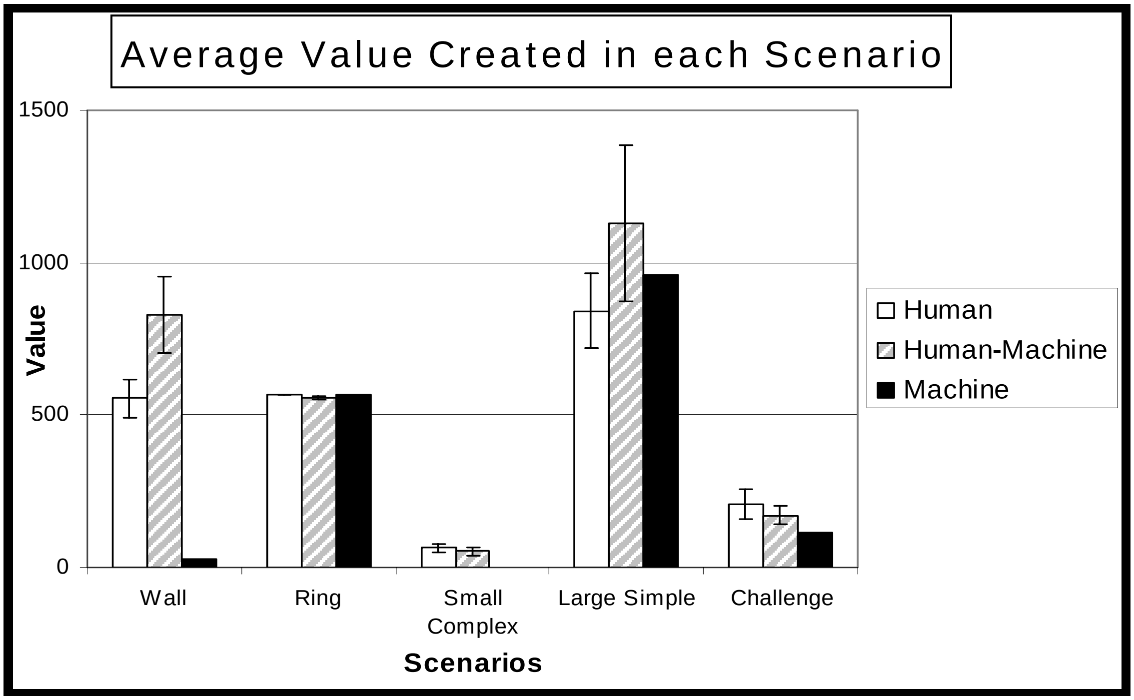


Figure 6-2: Average Value for Scenarios and Level of Automation

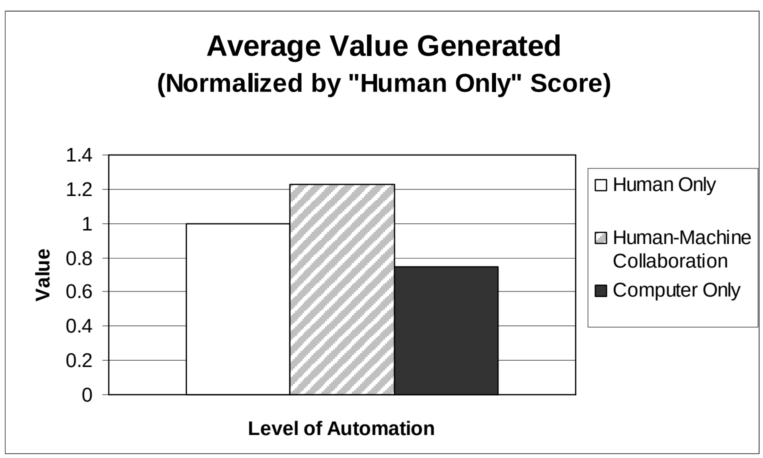


Figure 6-3: The Overall Average Value Generated

There is a trade-off associated with the increased value of the human machine collaboration plans. In addition to generating the most value, these plans also took the most amount of time to create. The additional time to create the collaborative plans can be attributed to the humans evaluating and studying the KPI in order to select cluster options (Step 9 in the HMCDM Experiment Algorithm, see Chapter 5). This step was not part of the decision making algorithm in either the human only or computer only experiments. Figure 6-4 shows the average time to create the initial plans for each level of human and machine interaction: human only – 102 seconds, human-machine collaboration – 289 seconds, and computer only – 81 seconds. Human-machine collaboration plans took a little over three additional minutes to create. This is not a significant issue because this additional time would be in the mission pre-planning phase. However, this finding has implications for real-time, dynamic human-machine collaborative re-planning in a time critical mission. In such a case, the additional three minutes could pose a problem. Figure 6-5 portrays the relationship between the time to create the plan and the value achieved. The graph suggests a direct correlation between the two factors.

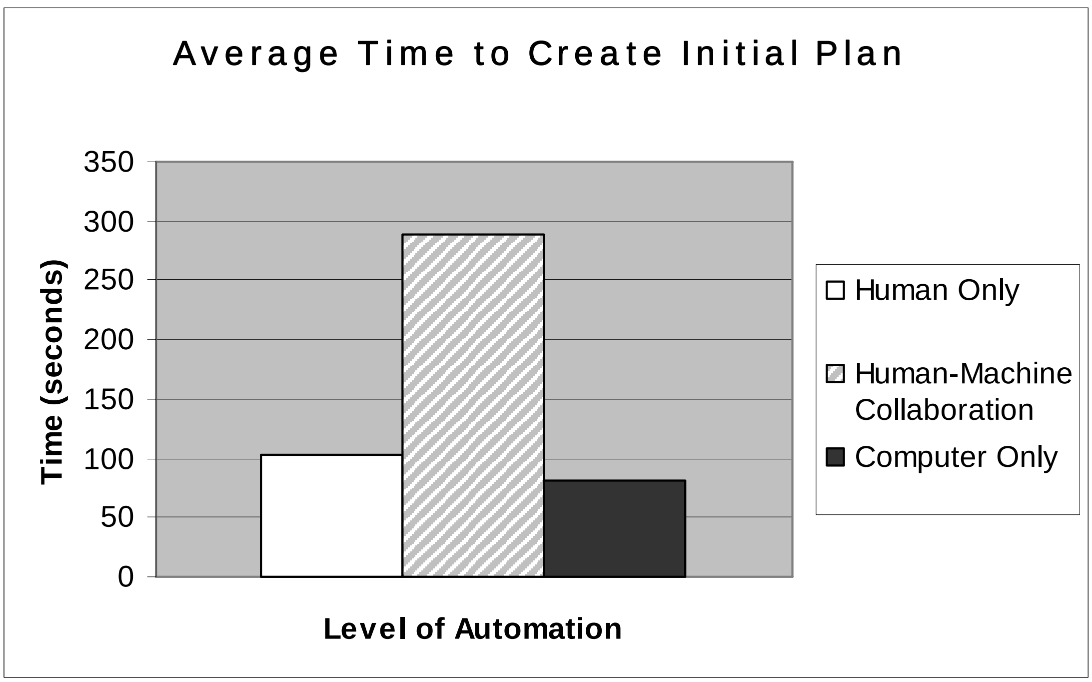


Figure 6-4: Average Time to Create Initial Plans - Across All Scenarios

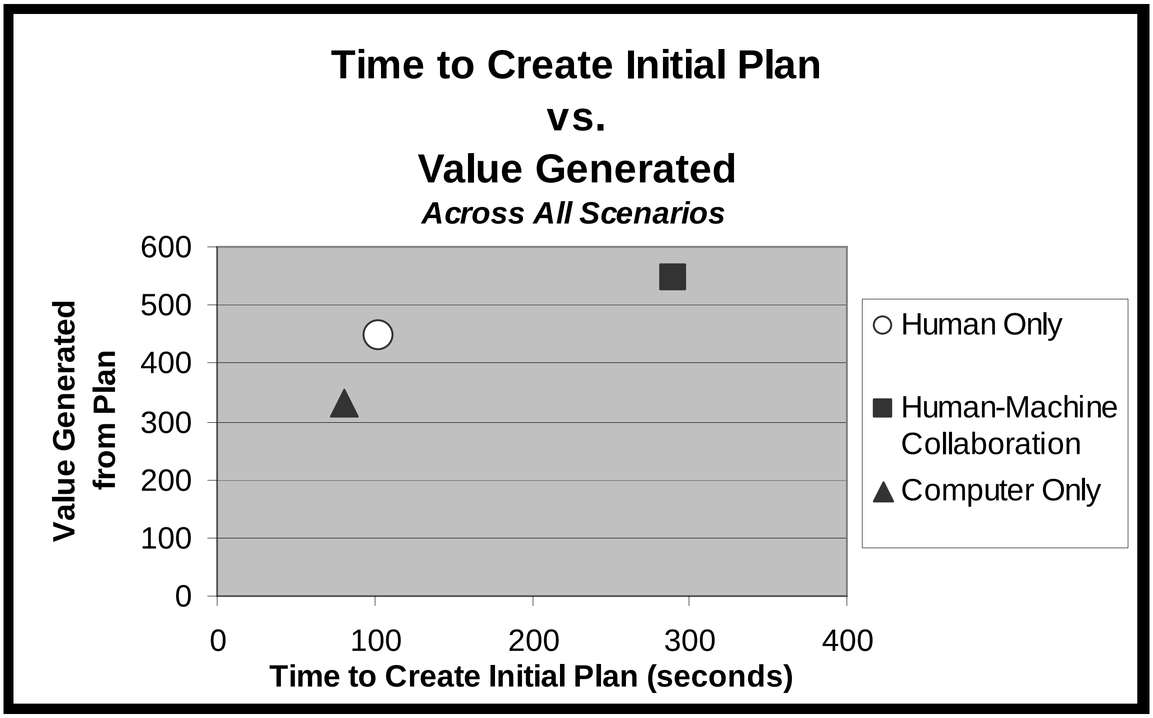


Figure 6-5: Value of Solution versus Time to Create Plan

During human-machine collaborative planning, the subjects were provided feedback in the form of KPI about the clusters they created and clusters that were generated by the computer (explained in detail in Chapter 5). One of the pieces of information was whether the cluster was created by the computer or the human. The subjects were then responsible for selecting clusters to place in the final solution. The subjects more frequently chose the cluster options they created, as shown in Figure 6-6. Human created clusters were chosen 60.6% of time while only 18.3% of the computer generated clusters were chosen. This was because the subjects understood the strategies they applied to construct their own clusters. The subjects were less likely to understand the computer generated solutions because they did not completely understand the rationale behind them. A detailed breakdown of the analysis for each of the five scenarios is given later in Section 6.7.4 and Figure 6-28.

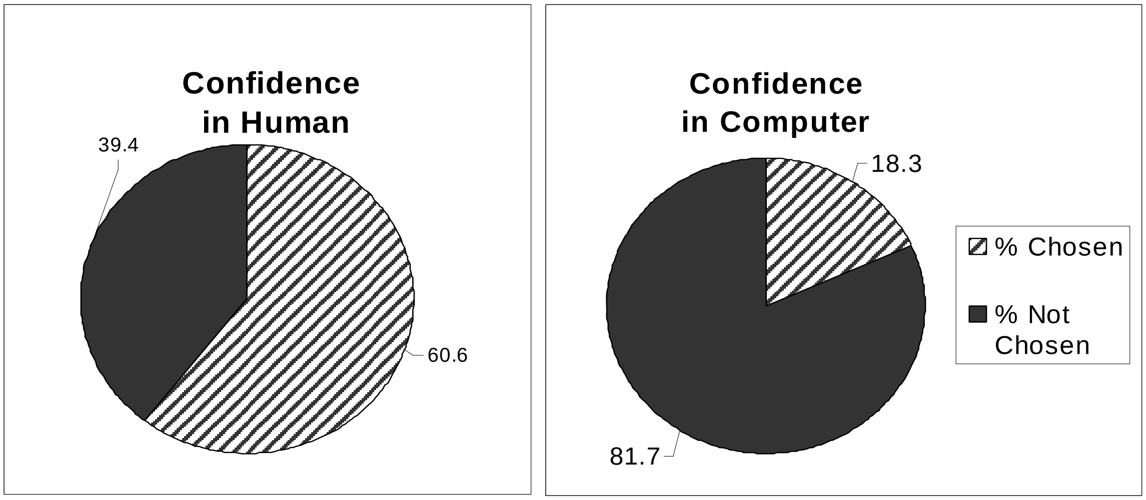


Figure 6-6: Percentage of Human and Computer Created Clusters Ultimately Chosen by the Human Participants for Inclusion in the Final Plan

6.2 Scenario #1 – Wall of Medium SAMs

This scenario contains a medium number of targets and is simple in complexity. A map layout depicting the spatial configuration is shown in Figure 6-7. The *Wall Scenario* consists of a "wall" of low value threatening targets (Medium SAM Sites with a value of 25 each), which block access to three very high value, low threat targets (TEL Supports with a value of 300 each).

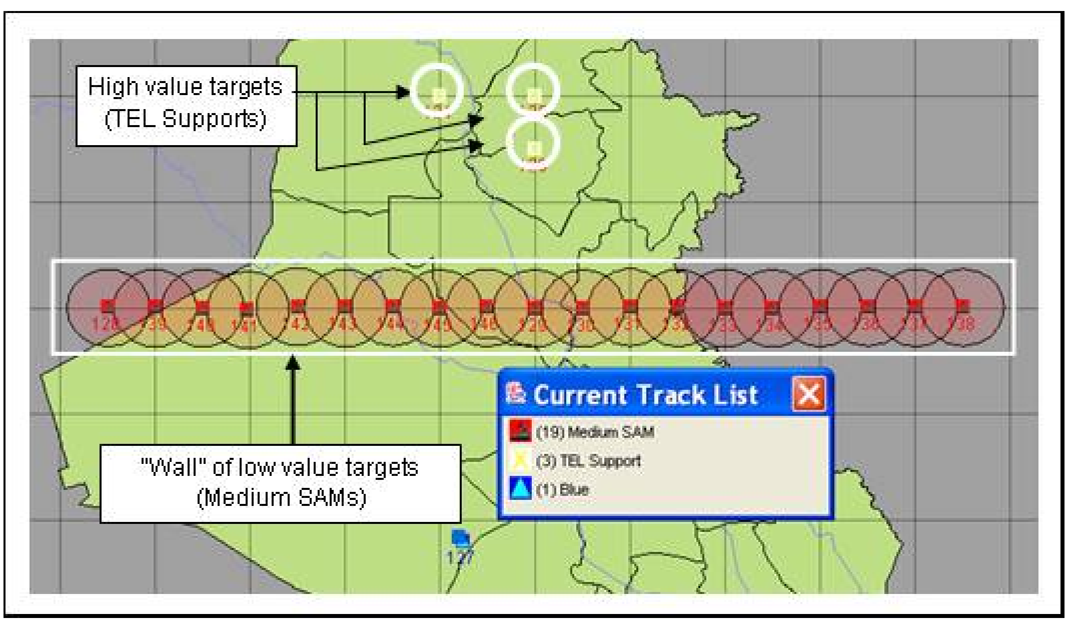


Figure 6-7: Map Layout of Wall Scenario

6.2.1 Results of Wall Scenario

Figure 6-8 shows that the results for this scenario are in favor of involving a human in the process of creating the enemy target clusters. The "Avg. Time to Create Initial Plan" is the only metric that favors using a computer to conduct the clustering. The metrics depicting the attrition of aircraft, number of aircraft used, and number of weapons loaded were relatively equivalent throughout each of the three levels of human-machine interaction. However, there is a significant disparity in terms of the value created. The average value created across the five participants for the "Human Only" level of involvement was 555, while the average for the "Human-Machine Collaboration" was 830. These numbers are in comparison to the value of 25, which was generated when the clustering was left to the computer alone. These results correspond to a 3220% increase in the value of the solution when human-machine collaboration was present versus the computer only solution. However, there is a trade-off for this significant increase in value. Solutions for this scenario involving human-machine collaboration took, on average, more than two minutes longer to create. The "Human Only" solution contains 2120% more value than the "Computer Only" solution but took an average of twenty four seconds more to create. In an operational setting, the time criticality of the mission would determine which level of human involvement was the best. If time was not a vital element to the mission, having a human and machine collaborate to create clusters would be the best course of action. However, if planning time was very critical, the computer might be the best option. Solutions generated by a human provide a good balance as they generate relatively high value at a fast time.

Wall Scenario Results

	Human Only Clustering	Human-Machine Collaboration	Computer Only Clustering
Avg. Time to Create Intial Plan (seconds)	50	165.8	26
Average Value Generated	555	830	25
Max Value Generated	675	1000	25
Avg. Attrition of Aircraft	3	3.4	3
Avg. # of Aircraft Initially Used	12.2	12	12
Avg. # of Weapons Loaded	125.6	128	128

Figure 6-8: Summary of Wall Scenario Results

6.2.2 Discussion

When the computer was responsible for clustering the enemy targets, it generated numerous small clusters of the low value threatening targets (Medium SAMs) and no clusters involving the high valued targets. Even though the TEL Support targets were worth considerably more, the computer could not find a safe and efficient way to attack these targets. The computer algorithm deems these high value targets as "unavailable" because they are blocked by the wall of targets. The high value targets are not clustered with any targets on the wall because the distance between the wall and the high value targets is very large. Also, because the threats do not physically overlap the high value targets, there are no explicit CaP constraints (see Section 3.2.3 for discussion on CaP constraints). The computer is limited by what the designers have programmed it to do, therefore in situations such as the Wall Scenario, the computer cannot generate the best solution. This occurs because the algorithm does not take into account the "big picture" of the scenario and has limited ability to reason spatially. In addition, humans can draw on past experiences and apply problem specific strategies to help with the clustering, while the computer cannot. Figure 6-9 illustrates the resulting plan when the clustering is handled exclusively by the computer.

In this particular scenario the human strengths of visual perception, pattern recognition, intuition, and strategic assessment (see Section 2.7.1.1 for a description of these strengths) allow the participants to understand that a plan can be created to "blow a hole" through the wall of threats and subsequently attack the high valued targets. This was accomplished by placing high value targets in the same cluster as a threat in the wall. By taking advantage of this ability to apply the appropriate "strategy," the humans were able to generate superior solutions. An example of one subject's human-machine collaboration experiment is shown in Figure 6-10.

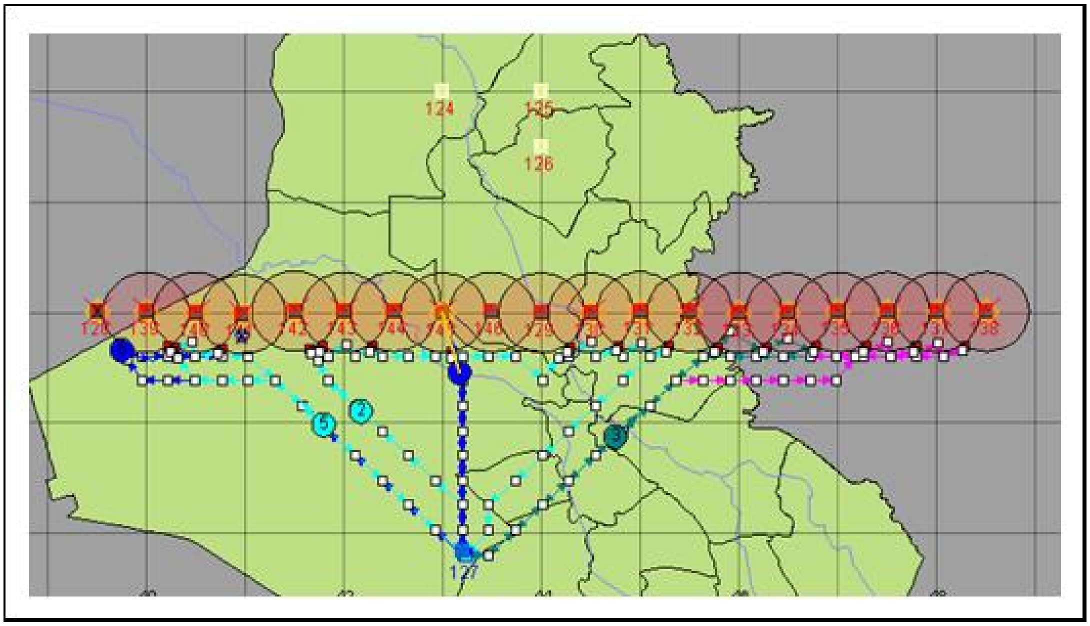


Figure 6-9: Result of "Computer Only" Clustering the Wall Scenario

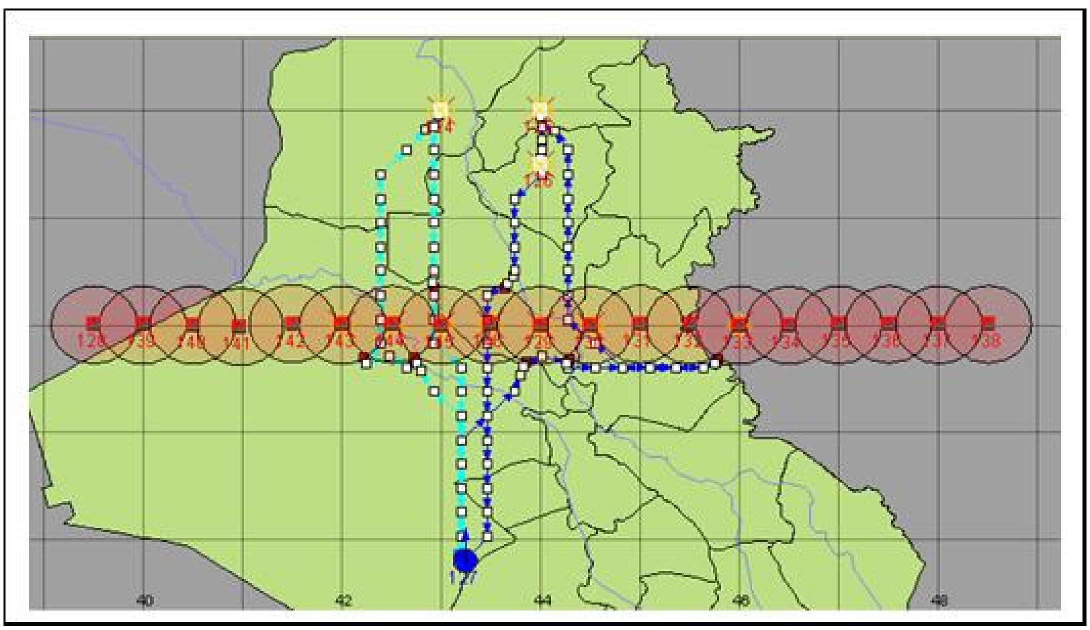


Figure 6-10: Result with Human Involved in Clustering the Wall Scenario

It is possible that new logic could be added to the existing computer clustering algorithm in order for it to create solutions similar to those generated with human involvement. However, for this to be effective, an exhaustive list of all intricacies in any possible scenario would have to be created and logic added to account for all of these potential scenario nuances. Instead of this time consuming design and development task, we can tap into human strengths to identify strategies quickly to overcome problem-specific aspects.

6.3 Scenario #2 – Ring of Medium SAMs

The *Ring Scenario* contains a medium number of targets and a medium amount of complexity. The complexity in this scenario arises from multiple coverage. A high value target (TEL Support with a value of 300) is covered by a low value target with a large threat radius (Long Launcher with a value of 25), which is itself covered by a combination of low value targets with small threat radii (Medium SAMs with a value of 25). Figure 6-11 provides a map layout of the scenario.

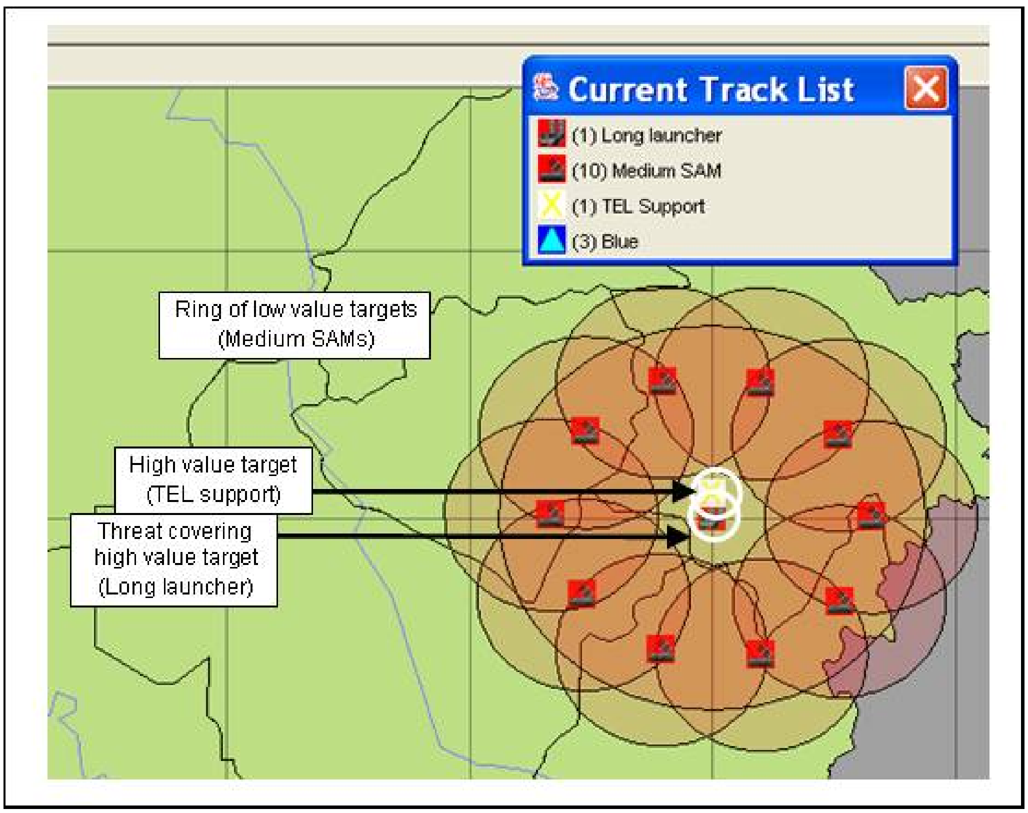


Figure 6-11: Map Layout of Ring Scenario

6.3.1 Results of Ring Scenario

Ring Scenario Results

	Human Only Clustering	Human-Machine Collaboration	Computer Only Clustering
Avg. Time to Create Intial Plan (seconds)	58.6	203.4	46
Avg. Simulation Time to Destroy TEL	2.67h	2.67h	3.11h
Average Value Generated	565	555	565
Max Value Generated	565	565	565
Avg. Attrition of Aircraft	0	0	0
Avg. # of Aircraft Initially Used	7.2	4.8	6
Avg. # of Weapons Loaded	67.2	57.6	72

Figure 6-12: Summary of Ring Scenario Results

The results of the Ring Scenario are shown in Figure 6-12. All three levels of automation generated quite similar values. The differences in the solutions arise in the time and resource usage metrics. The solutions with human involvement were able to attack the high valued target quicker and use fewer resources. The amount of time taken to attack the high value target is less with a human involved because the human has most likely focused their attention on this target. This results in the high value target being destroyed after an average of 2.67 hours in both the human only and human-machine collaboration experiments, while the computer only clustering took 3.11 hours to achieve the same result. Figure 6-13 shows the accumulation of value over time when the clustering was done by the computer and the results from one human participant. This graphic demonstrates that the human was able to generate the same amount of value but at a much quicker rate.

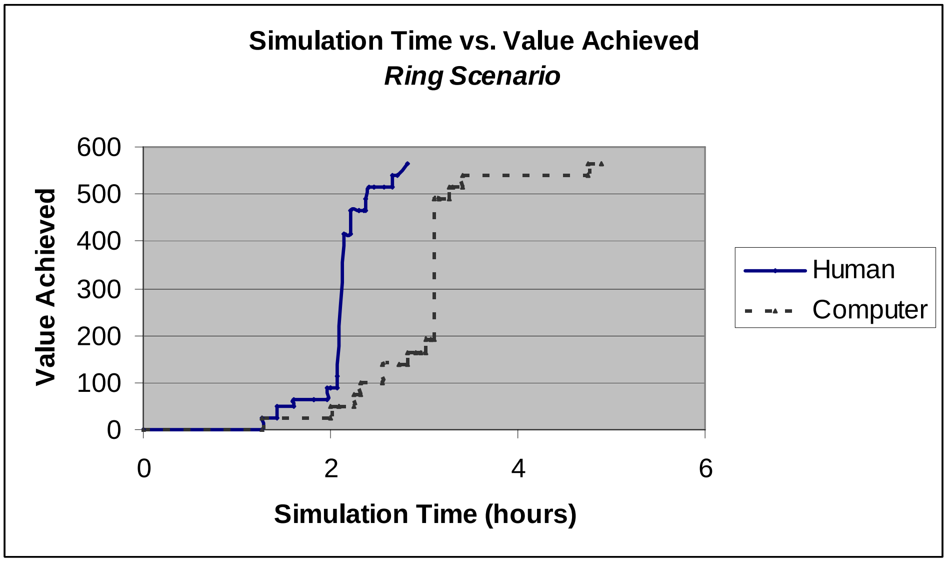


Figure 6-13: Simulation Time vs. Value Achieved in Ring Scenario

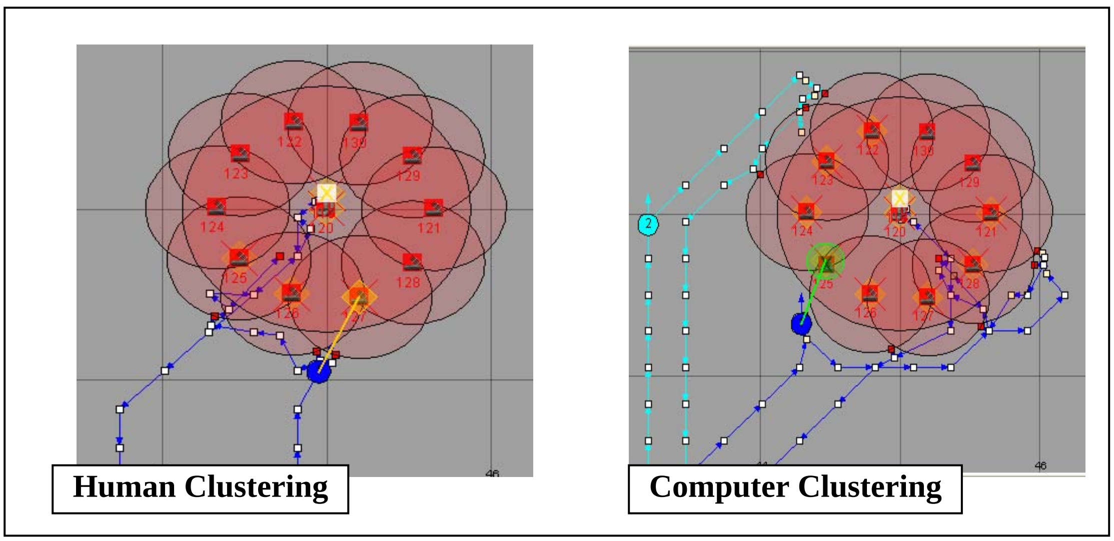


Figure 6-14: Human vs. Computer Clustering of Ring Scenario

Figure 6-14 illustrates the initial plans created with and without human participation in the clustering of enemy targets. The human was able to create a cluster where only a small number of targets would be destroyed before the high value target could be attacked.

6.4 *Scenario #3 – Small Complex Scenario*

The *Small Complex Scenario* includes a small number of targets but a high degree of complexity. The high complexity arises from several instances of multiple coverage. The scenario consists of five overlapping Long Launchers, each of which covers two Medium SAMs. All seven threat targets also cover a high valued TEL Support. Figure 6-15 illustrates the map layout of the scenario. This scenario was devised to test a human's ability to de-conflict complicated threat coverage schemes. As the figure shows, it is not visually obvious which threat ring corresponds to which Long Launcher. This setup negates a human's ability to identify "good" clusters quickly based on spatial reasoning alone. Therefore, the computer only solution was hypothesized to outperform the human only or human-computer solutions.

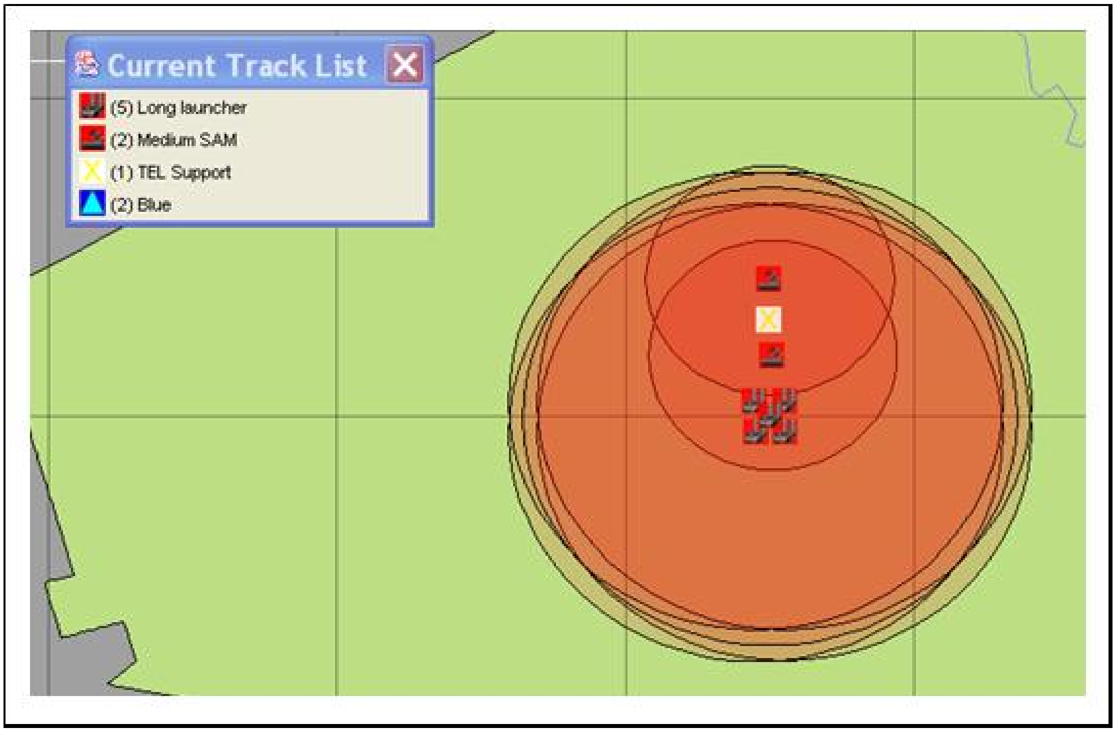


Figure 6-15: Map Layout of Small Complex Scenario

6.4.1 Results of Small Complex Scenario

The subjects employed two main clustering strategies in this scenario. The most common approach was to give up on determining which threat radius corresponded to which target and simply cluster all of the targets together. This tactic resulted in a total value of 40.

More ambitious subjects used the zooming function to de-conflict the overlapping targets. These subjects applied a more intelligent approach and broke the targets into a few small clusters. Those that used this strategy accumulated a value of 100.

6.4.2 Discussion

The results for this scenario were unexpected. The severe overlap of targets was designed specifically to confuse or overwhelm the human operator. It was hypothesized that the solutions created using computer only or human-machine collaboration levels of automation would dominate. However, Figure 6-16 shows that the best results were obtained when a human was exclusively in charge of creating the clusters of enemy targets. The solutions involving human and machine interaction contained slightly less value but also took twice the amount of time. The computer was not able to generate a plan that had a positive value and an acceptable amount of risk, therefore there is a column of zeros in Figure 6-16. In effect, the complication of the scenario actually confused the computer more than it confused the human subjects.

Small Complex Scenario Results

	Human Only Clustering	Human-Machine Collaboration	Computer Only Clustering
Avg. Time to Create Intial Plan (seconds)	100.2	203.6	36
Average Value Generated	64	52	0
Max Value Generated	100	100	0
Avg. Attrition of Aircraft	0.8	0	0
Avg. # of Aircraft Initially Used	6	6.2	0
Avg. # of Weapons Loaded	26.4	33.6	0

Figure 6-16: Summary of Small Complex Scenario Results

In this scenario, the original "gut feeling" of the human proved to be the best course of action. This is supported by the human only clusters, which resulted in the highest quality solutions. In the human-machine collaboration experiments, the additional Key Pieces of Information provided by the computer were enough to either mislead the human or shake their confidence in their original instinct. The participants' "gut feeling" resulted in better solutions than when they were given additional information to evaluate these solutions, suggesting that the additional information was actually detrimental to the human's decision making process. This outcome runs counter to the intuition that more information will lead to higher quality solutions.

6.5 *Scenario #4 – Large Simple Scenario*

The *Large Simple Scenario* entails a large number of targets and a low level of complexity. The scenario contains thirty-four targets that are distributed to allow the targets to be visually separated based on inspection alone. Even with the large number of targets, it was assumed that the humans would add significant benefit due to the visual groupings of targets.

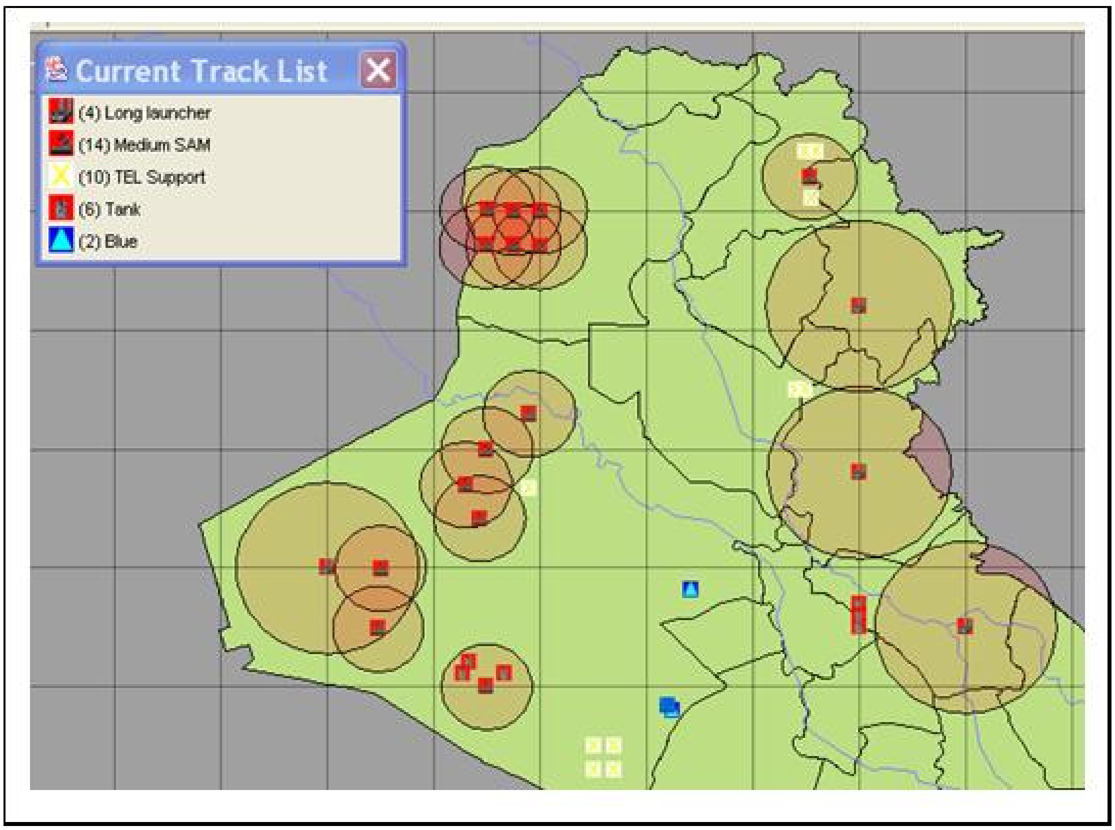


Figure 6-17: Map Layout of Large Simple Scenario

6.5.1 Results of Large Simple Scenario

Large Simple Scenario Results

	Human Only Clustering	Human-Machine Collaboration	Computer Only Clustering
Avg. Time to Create Intial Plan (seconds)	68.8	375.4	72
Average Value Generated	842	1130	959
Max Value Generated	1270	1845	959
Avg. Attrition of Aircraft	1.2	1.8	0
Avg. # of Aircraft Initially Used	9.6	9.8	9
Avg. # of Weapons Loaded	106	112.8	128

Figure 6-18: Summary of Large Simple Scenario Results

Figure 6-18 summarizes the results for this scenario. The experiments involving human-machine collaboration produced the best results for this scenario. These experiments, on average, generated the highest total value and had comparable numbers for the quantity of aircraft used and weapons loaded. However, the human participants did make an interesting trade-off between risk and reward. They appear to have been willing to accept more risk when there was a possibility for more value to be gained, portrayed by the higher attrition of aircraft in experiments with a human involved. This may be because the operators were working with unmanned aerial vehicles as their assets. They did not have to factor in the intangible cost of losing a human life. It is likely that if the friendly resources instead were manned vehicles, the humans would not have been willing to accept this additional risk.

6.5.2 Discussion

The main downside to the "Human-Machine Collaboration" experiments is the extensive amount of time taken to create the initial solutions. This is most likely attributed to the large number of targets in the scenario. The amount of targets led to both the human participants and the machine algorithm creating numerous total clusters. Once the users were presented with the KPI for each of the clusters, it took them a significant amount of time to study, evaluate, and select the desired clusters. For this scenario, the participants expressed concern that during human-machine collaboration, it was difficult to evaluate and select the options because there were so many different options to choose from. One of the main problems the participants expressed was ensuring that separate clusters containing the same target were not chosen. This concern with the number of options to evaluate ties back into the concept of human workload (see Evaluative Criteria in Section 2.5.1). These results suggest that achieving acceptable levels of human workload requires a delicate balance. An operator might become overwhelmed if there are excessive amounts of information or decisions to be made. In addition, the operator might become bored or too distracted if there is not enough work for them, leading to complacency or skill degradation [20].

6.6 Scenario #5 – Challenge Scenario

The *Challenge Scenario* was the final and most difficult scenario. The scenario is illustrated in Figure 6-19. It combined a large number of targets (*Large Simple Scenario*) with a high degree of complexity (*Small Complex Scenario*). There are a total of 192 enemy targets in the scenario and there is an enormous amount of complex coverage. The amount and degree of coverage is depicted with magnified views of the map layout in Figure 6-20. In this section of the scenario, there are twenty-four targets positioned in a very small amount of space. This arrangement makes it difficult for a human to visually interpret the relationships. A human is not able to distinguish between the targets until this portion of the map is magnified significantly. Once the zoom feature is used to recognize the targets adequately, it then becomes impossible to identify the range of the threat associated with each target.

6.6.1 Hypothesis

The humans were expected to be overwhelmed by the vast number of targets and varying degrees of threat coverage. As a result, they were expected to create a small number of clusters based on the few areas that might be easily attacked. In other words, within the large, complicated scenario, there are certain areas that a human could use spatial reasoning to identify clusters of targets quickly that could easily be attacked. Figure 6-21 shows an example of this concept. The targets that are identified by the arrows can be attacked with little to no risk while still generating a positive value of twenty-five points each to the overall solution.

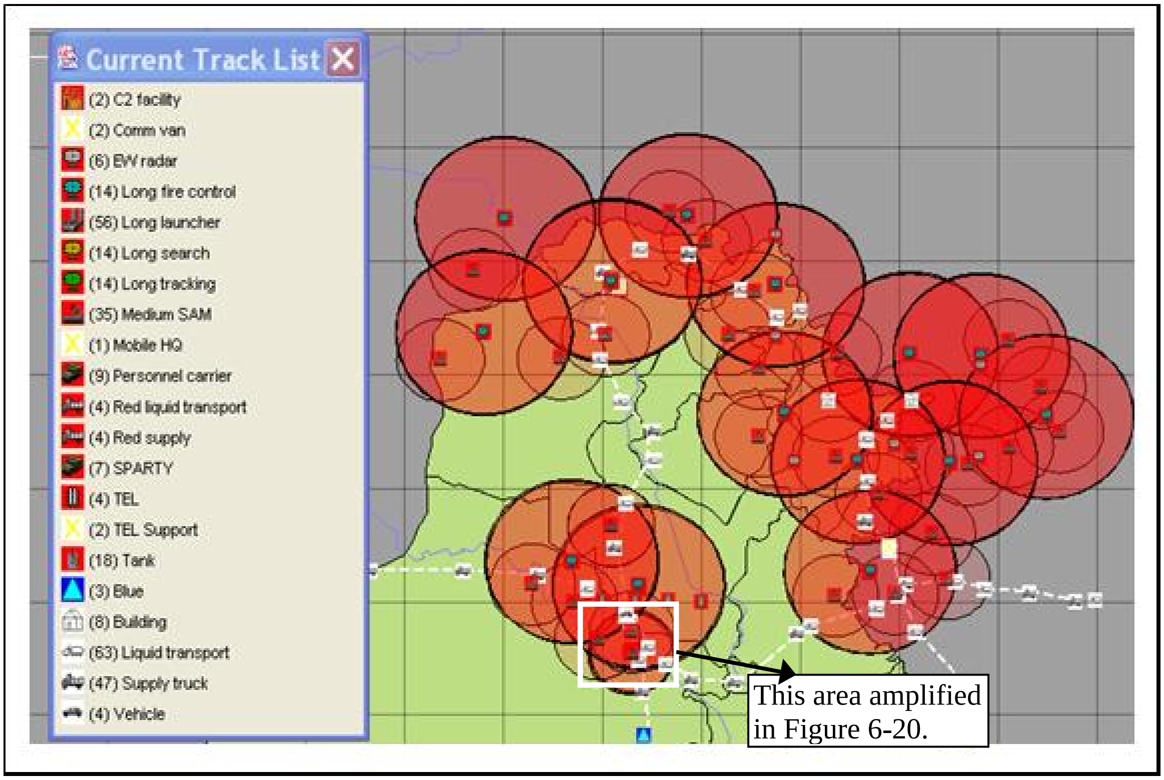


Figure 6-19: Map Layout of Challenge Scenario

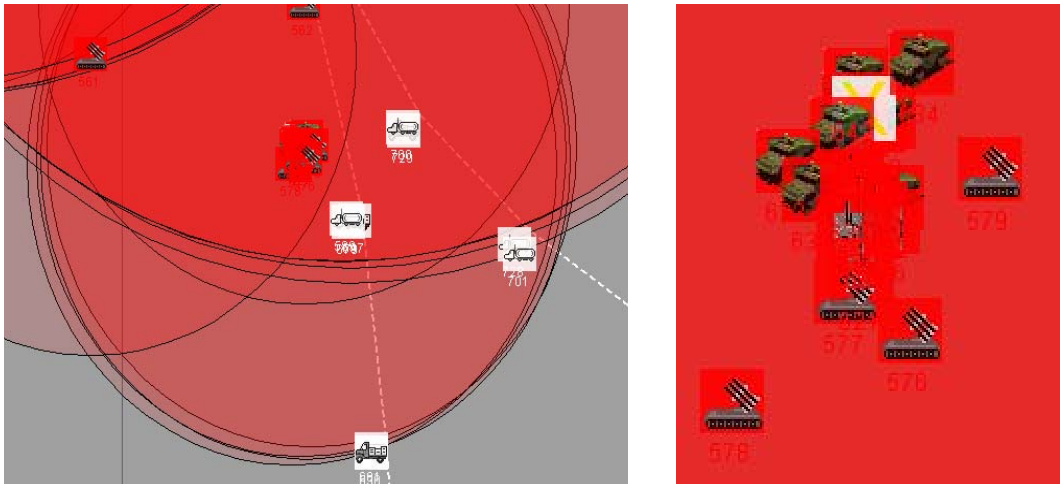


Figure 6-20: Magnified View of a Section in Map Layout (Shows the Difficulty in Distinguishing Threats)

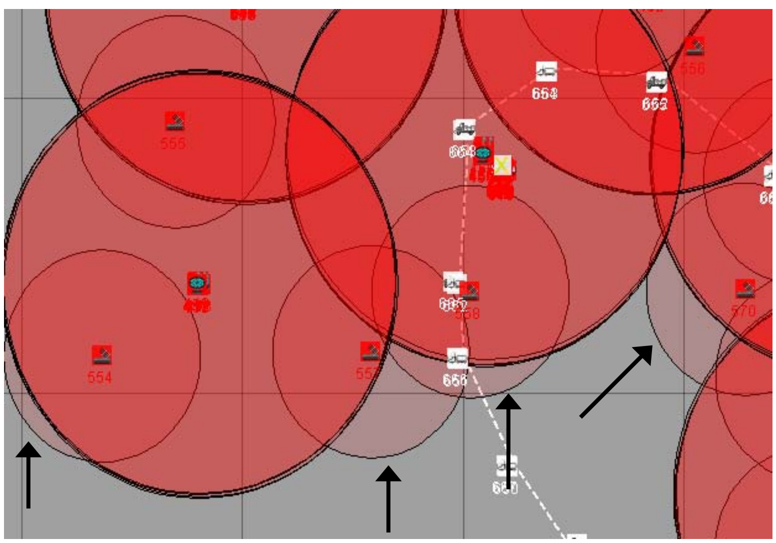


Figure 6-21: "Picking Off the Low Hanging Fruit"

6.6.2 Results of Challenge Scenario

The number of clusters created in the human only experiments was relatively small compared to the total number of targets contained within the scenario. There was an average of only 3.6 clusters created per human, each containing an average of 4.1 targets. This equates to only 14.76 (7.7%) out of the possible 192 targets being clustered in each of the experiments. We expected this outcome due to the extreme complexity in the scenario and the nature of the human only experiments. Experiments run at this level of automation gave participants only one shot at clustering the enemy targets. They had to base their decision solely on the map layout of the scenario. Therefore, the participants clustered very few targets in the human only experiments because there were not a lot of clusters that could be formed from spatial reasoning alone.

The results for the *Challenge Scenario* are summarized in Figure 6-22. The solutions created entirely by the computer generated the least amount of value, had the highest attrition of aircraft, and used the most resources (aircraft and weapons) out of the three levels of automation. Human-machine interaction generated a higher value, used the least amount of resources, and fell in the middle with regards to aircraft attrition. When clustering was left exclusively to the humans, the solutions generated the highest value and had the least number of aircraft lost. These solutions were created in approximately the same amount of time as the computer solutions. A drawback to the "Human Only" solutions is the number of resources used. On average, they used one more aircraft and had twenty-three more homing missiles loaded than HMCDM solutions. However, in practice, unless the resources were in a limited supply, it is likely that a commander would prefer the additional value, less risk, and shorter planning time in comparison to the HMCDM solutions.

Challenge Scenario Results

	Human Only Clustering	Human-Machine Collaboration	Computer Only Clustering
Avg. Time to Create Intial Plan (seconds)	233.4	499.4	225
Average Value Generated	207	170.4	115
Max Value Generated	400	270	115
Avg. Attrition of Aircraft	4.8	6	7
Avg. # of Aircraft Initially Used	12.2	11.2	13
Avg. # of Weapons Loaded	135.2	112	148

Figure 6-22: Summary of Challenge Scenario Results

6.6.3 Discussion

The human outperformed the human-machine collaboration in this scenario for the same reason as in the *Small Complex Scenario*. It turns out that the original intuition of the human operators proved to be the best course of action. Again, it is evident that the additional information provided in the KPI convinced the operators to change their minds about the compilation of the enemy target clusters. This outcome further supports the notion that more information does not always lead to higher quality solutions.

It is also important to note that mental workload issues played a role in this scenario as well. There was a significant amount of time between the humans entering clusters for more information and the actual KPI being calculated and displayed to the subjects. This additional time can be attributed to the great number of targets resulting in numerous calculations for the computer. The considerable break in action appeared to cause the users to become bored and lose their sharpness when selecting the options. Many of the participants had to re-familiarize themselves with their original inputted clusters before they started evaluating and selecting the clusters.

6.7 General Discussion

Overall, human-machine collaborative planning produced the best plans. The human strengths of pattern recognition, intuition, and spatial reasoning were combined with the computer strengths of data organization and fast calculation in order to create higher quality solutions. The additional time to create the human-machine plans can be attributed to the humans evaluating and studying the KPI produced by the computer to select cluster options. This step was not part of the decision making algorithm in either the human only or computer only experiments. However, in practice, the additional time expended is not an issue because it is only expended in the mission pre-planning phase. Therefore, the higher value and lower resource usage plans created by the human-machine collaboration are preferred.

6.7.1 Confidence and Satisfaction in Solutions

The added benefit from human-machine collaboration is not strictly confined to the quantitative increases in the values of the plans. Each of the experiment participants expressed much more confidence in the solutions created with machine collaboration. This was the case even in the scenarios in which their human only clustering outperformed the clusters created with collaboration. The additional feedback from the computer either solidified their stance on certain clusters or suggested alternative clusters that the human decided they liked better. They acknowledged having a better understanding of the solution that was going to be created compared to the human only approach and felt more "in-the-loop" than the computer only method.

6.7.2 Teams of Human Decision Makers

There is also evidence to suggest that solution quality could be enhanced by having teams of humans interact with the machine instead of single users. In many of the scenarios, there were certain subjects who would employ particular clustering strategies that resulted in superior results. Figure 6-23 and Figure 6-24 show that the same participants did not consistently generate the best results. There were many different top performers across each of the scenarios. This suggests that allowing teams of humans to collaborate with a machine might result in the best possible solutions being obtained. The highlighted entries in the figures correspond to the maximum value generated for each scenario

Participant Value Generated in HMCDM Experiments

Participant #	1	2	3	4	5
Wall Scenario	1000	325	950	950	925
Ring Scenario	565	540	565	540	565
Small Complex Scenario	40	100	40	40	40
Large Simple Scenario	960	375	945	1845	1525
Challenge Problem Scenario	97	125	190	170	270

Figure 6-23: Value Generated by each Participant in HMCDM Experiments

Participant Value Generated in Human Only Experiments					
Participant #	**1**	**2**	**3**	**4**	**5**
Wall Scenario	625	675	400	400	675
Ring Scenario	565	565	565	565	565
Small Complex Scenario	40	100	100	40	40
Large Simple Scenario	660	660	645	1270	975
Challenge Problem Scenario	400	130	205	115	185

Figure 6-24: Value Generated by each Participant in Human Only Experiments

Figure 6-25 shows the average value generated by the five participants for each scenario across the three levels of human-machine interaction using the human only score to normalize the values. This graphic shows that HMCDM created, on average, 64.5% more value than solutions created with the exclusive use of the computer. It also confirms that HMCDM solutions created 22.6% more value than human only solutions. On the other hand, Figure 6-26 shows the maximum value generated by one of the participants for each scenario across the three levels of interaction. Again, the human only score is used to normalize the values. This figure illustrates a much larger disparity between the values generated. The maximum value generated by one of the participants using HMCDM is 127% higher than the value created by the computer and 25.6% higher than the maximum human only value. It is possible that this additional value could be more consistently captured if the decisions were made by the collective team of participants instead of carrying out the decisions individually.

Summary Results Avg Scenario Value Generated (Normalized)	Human Only Clustering	Human-Machine Collaboration	Computer Only Clustering
Wall Scenario	1	1.495	0.045
Ring Scenario	1	0.982	1
Small Complex Scenario	1	0.813	0
Large Simple Scenario	1	1.342	1.139
Challenge Problem Scenario	1	0.823	0.556
Mean	**1**	**1.226**	**0.745**

Figure 6-25: Average Value Created in each Scenario: Normalized by "Human Only" Score

Summary Results Max Scenario Value Generated (Normalized)	Human Only Clustering	Human-Machine Collaboration	Computer Only Clustering
Wall Scenario	1	1.481	0.037
Ring Scenario	1	1	1
Small Complex Scenario	1	1	0
Large Simple Scenario	1	1.453	0.755
Challenge Problem Scenario	1	0.675	0.288
Mean	**1**	**1.256**	**0.440**

Figure 6-26: Maximum Value Created in each Scenario: Normalized by "Human Only" Score

6.7.3 Effects of the Graphical User Interface

The focus of this experiment was not on display design. However, the interface could be designed to mitigate some of the effects found in the human-machine collaboration experiments. There are many alterations to the GUI that could possibly increase the efficiency and effectiveness of the experiments. These adjustments are based on observations during the experiments as well as participant suggestions.

One design option would be the further development of the KPI display. In the GUIs' current form, the KPI are displayed through the use of an Excel spreadsheet (see Figure 5-11). This design was chosen to allow the users to compare each element of the KPI for the different options. However, this display could be improved to allow the operators to more easily compare de-conflicting options of clusters. The current design was limited for scenarios with a large number of targets. In these scenarios, the considerable number of targets resulted in many clusters containing the same targets. This led to the human participants taking a long time to study and select options. Figure 6-27 presents a breakdown of the amount of time spent on the KPI evaluation and option selection for each of the five scenarios. A design with a relatively quick and simple method for the users to identify if they are about to select multiple options that contain the same target would most likely reduce the time to select options.

Participants also spent a considerable amount of time trying to relate the options in the spreadsheet view and their corresponding locations on the map. A possible solution to this problem would be to provide a direct link between the displayed KPI for each option and the map layout of the targets. The users should be able to select an option and have all of the targets within this option become highlighted on the map view. This could be done by changing the color of the associated targets within the selected option.

Average Time for Participants to Study and Select Options Based on Given KPI

Wall Scenario	115 seconds
Ring Scenario	144 seconds
Small Complex Scenario	103 seconds
Large Simple Scenario	306 seconds
Challenge Problem Scenario	266 seconds

Figure 6-27: Time to Select Options in Human-Machine Collaboration Experiments

6.7.4 Option Selection

In the beginning of Chapter 6, it was mentioned that the human participants ultimately selected 60.6% of the clusters they created while only choosing 18.3% of the computer generated clusters. Figure 6-28 provides additional information, detailing the percentage of clusters selected in each of the five scenarios. The data suggests that the human participants are less likely to select their own clusters in situations containing a large number of targets. This is evidenced by the small difference in the percentage of clusters selected in the *Large Simple* and *Challenge Problem* scenarios. A somewhat unexpected result is that it appears the complexity in the scenario does not have a negative effect on humans selecting their own clusters. Aside from the *Challenge Problem*, the *Small Complex* and *Ring* scenarios were the only other scenarios with a medium or high level of complexity. In both of these scenarios, there was actually a large differential in favor of selecting human created clusters. It may be that in complex scenarios, the computer did not provide enough information in the KPI for the human to fully trust the computer generated options. This phenomenon would then have caused the humans to rely on the clusters they created themselves, which they understand better.

Summary Results

Trust in Human vs. Machine Created Clusters

	% of Human Created Clusters Chose by Human	% of Computer Created Clusters Chose by Human	Difference
Wall Scenario	77.8%	19.4%	**58.4%**
Ring Scenario	90.0%	10.0%	**80.0%**
Small Complex Scenario	83.3%	0.0%	**83.3%**
Large Simple Scenario	52.2%	13.8%	**38.4%**
Challenge Problem Scenario	38.9%	25.0%	**13.9%**
Totals	**60.6%**	**18.3%**	**42.3%**

Figure 6-28: Percentage of Human and Computer Created Clusters Ultimately Chosen by the Human Participants for Inclusion in the Final Plan in each Scenario

6.7.5 Quality of Options vs. Number of Options

We also studied the relationship between the value generated in the solution and the number of options the subject had to choose from in the human-machine collaboration experiments. Figure 6-29, Figure 6-30, Figure 6-31, Figure 6-32, and Figure 6-33 depict the results from each of the five scenarios. These results suggest that the key to creating solutions with higher value is not the *number* of options, rather it is the *quality* of options to choose from. Many of the participants expressed difficulty evaluating and selecting options when there were a large number of options. The only scenario with an apparent linear relationship between the number of options and value generated was the *Small Complex Scenario* (see Figure 6-31). This scenario also contained the least number of options for the subjects to select from.

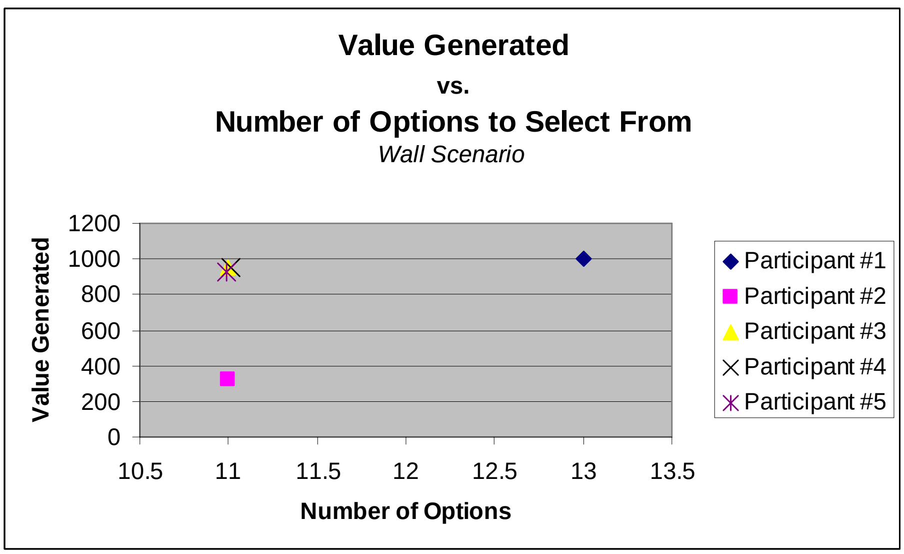


Figure 6-29: Comparison of Value Generated and Number of Options Wall Scenario

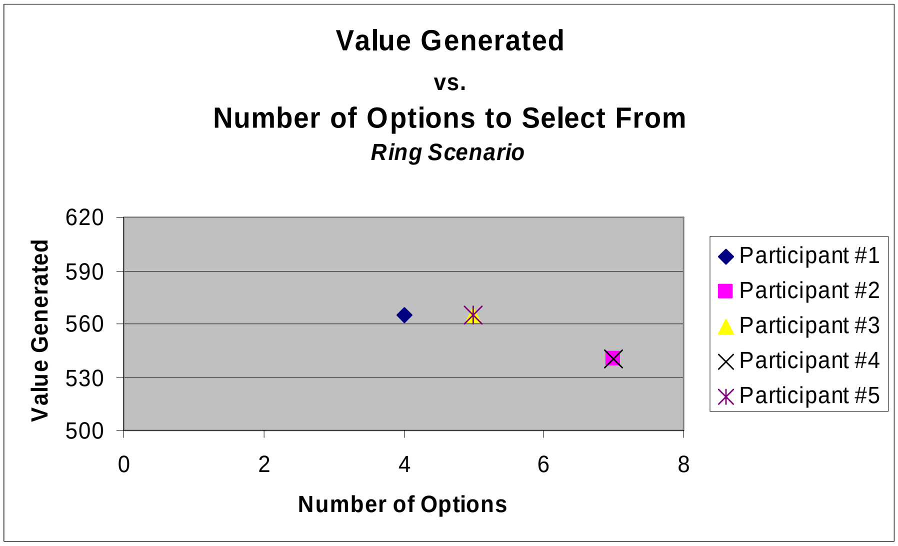


Figure 6-30: Comparison of Value Generated and Number of Options Ring Scenario

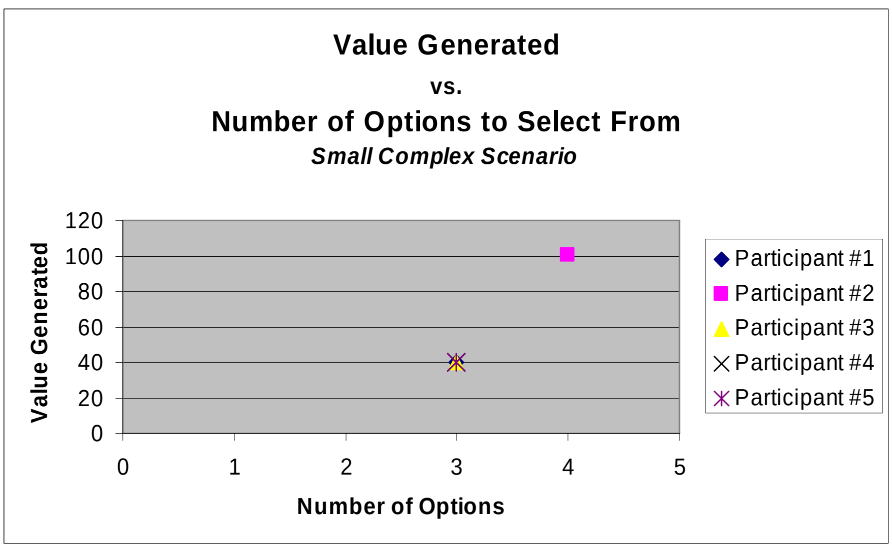


Figure 6-31: Comparison of Value Generated and Number of Options Small Complex Scenario

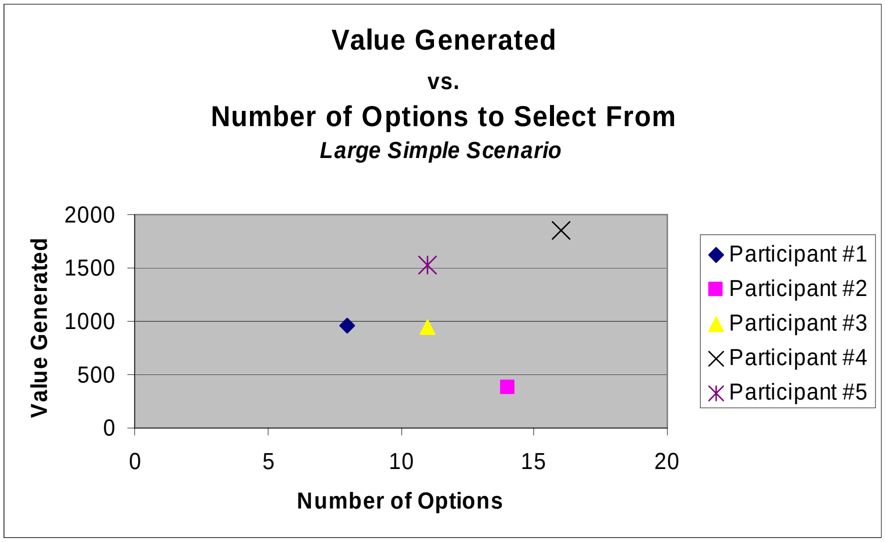


Figure 6-32: Comparison of Value Generated and Number of Options Large Simple Scenario

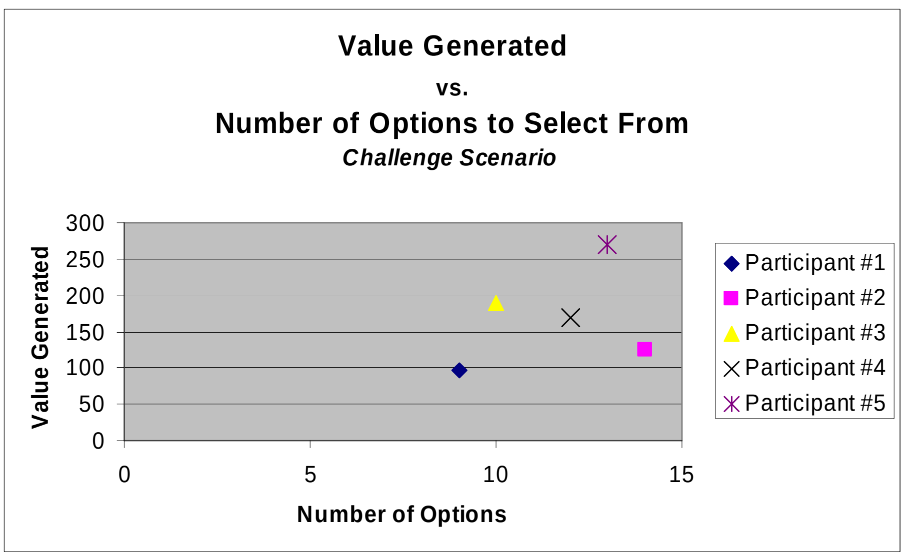


Figure 6-33: Comparison of Value Generated and Number of Options Challenge Scenario

6.7.6 Human and Computer Thought Processes

We also examined the amount of overlap in the clusters created by a human versus those by the computer. This would give us some insight into how similar the two were "thinking." A large amount of overlap would imply that the benefit from inserting a human-in-the-loop was minimal. Figure 6-34 shows that an extremely low number of identical clusters were created by both the human and computer. In fact, identical clusters were only created in one of the five scenarios and, even in that case, the average was less than one matching cluster per subject. The diverse clusters created by the humans and machines stress the importance for human-machine collaboration in order to produce the broadest range of clusters.

Human-Machine
"Similar Thinking"?

	Avg # of Identical Clusters Generated
Wall Scenario	0
Ring Scenario	0
Small Complex Scenario	0.8
Large Simple Scenario	0
Challenge Problem Scenario	0

Figure 6-34: Avg. Number of Identical Clusters Created in Each Scenario

6.7.7 Effect on Repeatability and Predictability

A potential drawback of human involvement is the impact on repeatability or predictability. Regardless of the number of times a specific scenario is run, the computer will always generate the same plan. The involvement of a human in the decision making process removes this predictability in the planning process. Each user has a different tolerance for risk and a different outlook on the trade-off between factors such as time, resource usage, risk, and value generated. For example, some humans might be willing to subject the UAVs to a large amount of risk because there are no human lives directly at stake while others are not willing to do so even if there is a potential for a large amount of value to be gained.

Chapter 7

Summary & Future Work

This research has focused upon the application of using HMCDM in a large-scale, complex optimization problem in an effort to generate more valuable solutions more quickly. This chapter serves as a summary of the work presented in this thesis as well as offering suggestions for future research.

7.1 Summary

Typical human-machine approaches start with a human process and augment it with decision-support, or start with an automated process and augment it with operator input. We provided an alternative to these approaches by presenting an HMCDM methodology that addressed collaboration from the outset of the decision-making design process. We updated and built upon previously accepted lists of human and computer strengths and capabilities. We built upon previous research to propose a methodology for determining the optimal level of automation when allocating decisions in a system or algorithm. We introduced a method for combining traditional goal decomposition with composite variable formulation into an *Iterative Composite Variable Approach* for solving large-scale optimization problems. We applied HMCDM and an introductory version of the *Iterative Composite Variable Approach* to a complex military resource allocation and planning problem and showed through experimentation the potential for improvement in the quality and speed of solutions.

In conclusion, our results suggest that it is possible to combine the strengths of a human and a computer synergistically to create better solutions to a large-scale, complex optimization problem (specifically a resource allocation and planning problem) than those that either could produce alone. Future C^2 planning systems can be improved if the humans and machines are fully integrated in a way that takes advantage of the strengths of both.

7.2 Future Work

In this section we provide suggestions for future research in applying HMCDM in large-scale, complex optimization problems.

ADDITIONS IN THE SPECIFIC RESOURCE ALLOCATION AND PLANNING PROBLEM

There are numerous logical extensions to the amount and type of human involvement in the resource allocation and planning system explored in this thesis, particularly in the creation of the composite variables. In addition to having the human involved with the creation of the clusters during the initial planning cycle, it might also be beneficial to allow the human to participate in this activity during each of the re-planning periods. Other possibilities include performing HMCDM in the sequencing of targets, creation of aircraft teams, and routing of individual aircraft. Discussion of these areas are given at the end of Chapter 3. In order to determine which of these options for human involvement add value to C^2 planning and resource allocation, further experiments would need to be conducted with different combinations of the possibilities described above.

MORE ITERATIONS IN ITERATIVE COMPOSITE VARIABLE APPROACH

The human machine collaboration experiments conducted for this research were a first attempt at applying HMCDM to an *Iterative Composite Variable Approach*. In the experiments, only one iteration was performed with HMCDM in the creation and updating of the composite variables contained within the pool of composites. Future research could investigate the effects of additional iterations.

FURTHER DEVELOPMENT OF KPI INTO SENSITIVITIES

The HMCDM experiments in this thesis provided human test subjects with Key Pieces of Information about the initial set of composite variables to test if a human could process this information and draw conclusions about how to change or alter the composites within the composite pool to generate a better solution. These KPI are ***not*** traditional sensitivities in that the subjects were not explicitly informed how the altering of the composites would affect the overall solution. We provided information which we deemed important (KPI) and relied on the human to draw conclusions about how the composites might be altered in an effort to produce a better solution. If instead, true sensitivities were provided to the human, the combination of HMCDM and an *Iterative Composite Variable Approach* might prove to be even more beneficial in solving large-scale, complex optimization problems.

BETTER USER INTERFACE

Enhancements in the graphical user interface might allow for more intelligent or efficient methods of combining human and computer strengths into a HMCDM process. In particular, the display of KPI or sensitivities could be improved.

ALTER THE METHOD OF HUMAN-MACHINE COLLABORATION IN CREATION OF COMPOSITES

The current method for human-machine interaction in our research resulted in the computer creating composites (clusters) independently using its own algorithm. An enhancement of this method would be to alter the computers process for creating its initial set of composites. One idea is to have the computer attempt to create “similar” clusters to those created by the human. If a computer were to be able to understand why a human considered their own clusters *good*, they might be able to intelligently perturb the human generated clusters and offer more *good* composites to the composite pool.

[This Page Intentionally Left Blank]

References

[1] Adams, Milt. "Mixed Initiative Planning and Control Under Uncertainty Design Document." Technical Report CSDL-R-2934, C.S. Draper Laboratory, January 2004.

[2] Adams, Milt. "Mixed Initiative Planning and Control Under Uncertainty Final Report." Technical Report CSDL-R-2978, C.S. Draper Laboratory, January 2004.

[3] Adams, M. and S. Kolitz, "Hierarchical Rail Traffic Flow Control", Proceedings of COMPRAIL 92 Conference, Washington DC, August, 1992.

[4] Anderson, D., Anderson, E., Lesh, N.B., Marks, J.W, Mirtich, B., Ratajczack, D.; Ryall, K., "Human-Guided Simple Search", *National Conference on Artificial Intelligence (AAAI)*, August 2000.

[5] Armacost, A., "Composite Variable Formulations for Express Shipment Service Network Design." PhD Thesis in Operations Research. Massachusetts Institute of Technology, Cambridge, MA, September 2000.

[6] Armacost, A.P; Barnhart, C.; Ware, K.A and Wilson, A.M, "UPS Optimizes Its Air Network", *Interfaces*, 34 (1), pp. 15-25, January 2004.

[7] Barnhart, C., and Cohn, A., "Composite-Variable Modeling for Service Parts Logistics," submitted to Annals of Operations Research, Special Issue on Transportation and Logistics Optimization Models and Solution Algorithms, January 2002.

[8] Barth, C., "Composite Mission Variable Formulation for Real-Time Mission Planning." Masters Thesis in Operations Research. Massachusetts Institute of Technology, Cambridge, MA, June 2001.

[9] Bertsimas, D. and Tsitsiklis, J.N. Introduction to Linear Optimization. Athena Scientific, Belmont, MA, 1997.

[10] Caballero, R.; Gomez, T.; Luque, M.; Miguel, F.; Ruiz, F., "Hierarchial Generation of Pareto Optimal Solutions in Large-Scale Multiobjective Systems." *Computers and Operations Research*, pp. 1537-1558, September 2002.

[11] CADRC, Cal Poly, San Luis Obispo: IMMACCS – Background Theory (Jens Pohl, February 1999) *Newsback.*

[12] Chien, S., Rabideau, G., Willis, J., and Mann, T. "Automating planning and scheduling of shuttle payload operations." *J. of Artificial Intelligence*, 114, pp. 239 – 253. 1999.

[13] Clark, S.J., "Large-Scale Optimization Planning Methods for Distribution and Transportation Logistics: An Application to Contingency Munitions." Masters Thesis in Operations Research. Massachusetts Institute of Technology, Cambridge, MA, June 2001

[14] Colgan, L., Spence, R., and Rankin, P. "The Cockpit Metaphor." *Behaviour & Information Technology*, 14(4), pp. 251 – 263. 1995.

[15] Fitts, P.M.. Human engineering for an effective air navigation and traffic control system. Washington D.C.: National Research Council. 1951.

[16] Hillier, F.S and Lieberman, G.J. "Introduction to Operations Research." McGraw-Hill, New York, 2001.

[17] Jakkola, Tommi S. MIT AI Lab. Lecture 16: Machine Learning Slides.

[18] Kantowitz, B.H. and Sorkin, R.D.. Allocation of functions. In: G. Salvendy (Ed.), Handbook of human factors, (pp. 335-369). New York: John Wiley & Sons, Inc. 1987.

[19] Kochhar, S. and Friedell, M. "User control in cooperative computer-aided design." *Proc. Of UIST '90*, pp. 143-151.

[20] Lee, John D. and See, Katrina A. "Trust in Automation: Designing for Appropriate Reliance." Human Factors, Vol. 46, No. 1. 2004.

[21] McCarthy, J. "Applications of Circumscription to Formalizing Common Sense Knowledge." *Artificial Intelligence*, 28, pp 89-116. 1986.

[22] Meister, D. Behavior Analysis and Measurement Methods. New York: John Wiley & Sons Inc. 1985.

[23] Neilson, C.A. "Large-Scale Network Design using Composite Variables: An Application to Air Mobility Command's 30-day Channel Route Network." Masters Thesis in Operations Research. Massachusetts Institute of Technology, Cambridge, MA, June 2002.

[24] Nemhauser, G. and Wolsey, L., *Integer and Combinatorial Optimization.* John Wiley and Sons, New York, 1999.

[25] Price, H.E. "The allocation of functions in systems." *Human Factors*. Vol. 27, No. 1, pp. 33-45. 1985.

[26] Roth, E.M.; Hanson, M.L; Hopkins, C.; Mancuso, V.; Zacharias, G.L. "Human In The Loop Evaluation of a Mixed-Initiative System for Planning and Control of Multiple UAV Teams." *Human Factors and Ergonomics Society 48th Annual Meeting*. 2004.

[27] Scott, S.D.; Lesh, N.; Klau, G.W., "Investigating Human-Computer Optimization", *ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 155-162, April 2002.

[28] Sheridan, T.B., R. Parasuraman, and C.D. Wickens, "A model for types and levels of human interaction with automation," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 30, No. 3, 2000.

[29] Sheridan, T.B. and Verplank, W., "Human and Computer Control of Undersea Teleoperators." Cambridge, MA: Man-Machine Systems Laboratory, Department of Mechanical Engineering, MIT. 1978.

[30] Sims, K. "Artificial evolution for computer graphics." *Computer Graphics*, 25(3) (*Proc. Of SIGGRAPH '91*), pp. 319-328. 1991.

[31] Todd, S. and Latham, W. Evolutionary Art and Computers. Academic Press. 1992.

[32] US Department of Defense. Human engineering procedures guide. Washington D.C: DoD-HDBK-763. 1987.

[33] Waters, C.D.J. "Interactive vehicle routing. *J. of the Operational Research Society*, 35(9), pp. 821-826. 1984.